

Radio Elettronica & Computer

11 programmi
per C64
e C128

Anno XVIII - N.8 - Ottobre 1989 - L. 8.500

Tassa pagata per campione allegato

UTILITY
FORMATTORE DI CASSETTE

ASSEMBLER
GESTIONE DEL RASTER

ESPANSIONE GRAFICA E I/O

TIPS & TRICKS
EFFETTI SPECIALI E SCROLLING

AMIGA
TAVOLA GRAFICA, SCHEDE XT/AT
E 68020

Trasferimento
automatico
dei programmi
da cassetta a disco



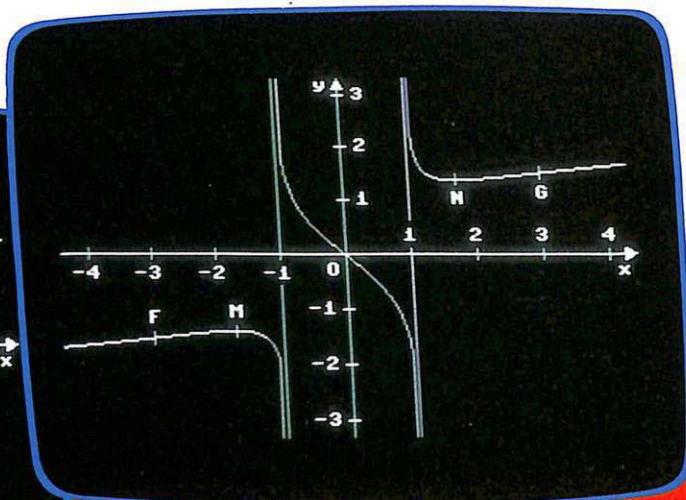
CONTABILITÀ
Contratto d'affitto
non abitativo

Tutto COMMODORE

Anno II - Numero 27 - Ottobre 1989 - L.13.000

**IMPARA LA
MATEMATICA
CON IL C 64**

Matematica



TUTORIAL

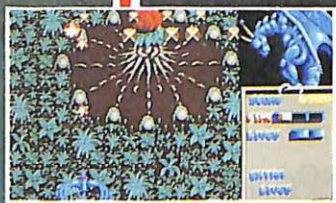
**Risoluzione
automatica di
problemi**

ESERCIZI

**Problemi svolti
e test sul
calcolo
integrale**

Gruppo Editoriale
JCE

è in edicola



DRAGON SPIRIT™



Scorch through the skies in the most challenging flight of your life.
Breath fire over hordes of attacking creatures from a prehistoric age.

- A scorching, rip-roaring fight to the death
- Pick up bonuses for mega weapons and extra firepower
- 8 levels of pulse-racing action
- Exhilarating and challenging...
- Have you got the fighting spirit? Have you got Dragon Spirit?

Programmed by: Consult Software Ltd
© 1989 TENGGEN INC. All Rights Reserved
TM and © 1987 NAMCO LTD

TENGGEN

The New Name In Coin-Op Conversions.

DOMARK

LEADER

DISTRIBUTION

Rubriche:

Software news pag. 6

Posta pag. 66

12 TAVOLA, PENNA E CALAMAIO

La vita degli appassionati e dei professionisti in Computer Art diviene di giorno in giorno più facile grazie alle rivoluzionarie innovazioni tecniche. Questo mese vi presentiamo un sofisticatissimo accessorio per disegnare su Amiga, un utile strumento per fare facilmente grafica anche con il Commodore 64 e una interessante rassegna di accessori vari

18 LOGO E GRAMMATICA



Continua il graditissimo corso di programmazione Logo. In questa puntata, articolata in sei paragrafi, vedremo alcune regole sintattiche che caratterizzano il linguaggio Logo e la trattazione di parole e liste

31 COME AFFITTARE PER NON ABITARE



Sul numero scorso abbiamo visto come mettere su carta la versione definitiva di contratti a uso abitativo con l'aiuto di una qualsiasi stampante. Chiudendo l'argomento, questo mese pubblichiamo un programma anche per contratti a uso non abitativo per coprire tutte le esigenze dei nostri affezionati lettori

37 CHI NON CERCA... TROVA LO STESSO



Da oggi anche gli utenti in possesso del solo registratore a cassette potranno automatizzare la ricerca dei programmi come è possibile fare solo usando un drive. In una manciata di linee Basic abbiamo racchiuso il sistema per costringere il registratore a cercare alla massima velocità i vostri programmi sulla cassetta

43 UNA SPREMUTA DI SPRITE



Iniziamo questa attesissima rubrica analizzando la struttura di un programma che gestisce in modo decisamente avanzato gli sprite hardware del C64, spremendo e usando al limite tutte le potenzialità nascoste del Vic II e della arcana struttura di funzionamento del Raster. Sul prossimo numero vedremo, in modo approfondito, come sfruttarne le funzioni

48 GRAFICA A BASSA RISOLUZIONE E DRIVE



Ecco chi sono i protagonisti di questa puntata dedicata alla mega espansione Ultra Basic. Creare finestre e implementare la tecnica degli overlay non sarà più un problema: poche e semplici istruzioni faranno per voi tutto il lavoro necessario

52 NEL VIVO DEL LINGUAGGIO

Iniziamo finalmente ad addentrarci nel vivo del linguaggio. In questa puntata sono di scena infatti i tipi predefiniti, le variabili e gli operatori di Qpl. Con questi elementi potete gustare le prime soddisfazioni di chi impara un nuovo linguaggio

Sommario

58 UN KIT PER MOZZARE IL FIATO



Davanti agli schermi dei computer qualche volta si rimane a bocca aperta e senza fiato. Anche il Commodore 64 ha questo fantastico potere nei confronti dell'osservatore e i tips di questo mese ne sono la dimostrazione

61 COME TU VUOI

Ecco una panoramica sugli accessori che permettono all'Amiga 2000 di concorrere con tutti i più diffusi e potenti personal computer sul mercato. Parliamo di schede interne, per emulare Ibm e per dare sprint all'elaborazione di Amiga, e di una stampante di grande successo

Caricate così i programmi della cassetta allegata:

Riavvolgete il nastro e digitate **LOAD** seguito da **RETURN** sulla tastiera del C64 e **PLAY** sul registratore. Verrà caricato il programma di presentazione con il menu dei programmi. Digitate **RUN** seguito dalla pressione del tasto **RETURN**. Terminata la presentazione, per caricare uno qualsiasi dei programmi è sufficiente digitare: **LOAD "NOME PROGRAMMA"** seguito dalla pressione del tasto **RETURN**.

Direttore Editoriale
Area Informatica
Marinella Zetti

Direttore responsabile
Paolo Romani

Caporedattore
Fernando Zanini

Responsabile grafico
Desktop Publishing
Adelio Barcella

Impaginazione elettronica
Denise De Matteis

Segretaria di redazione
Alessandra Marini

Collaboratori
Paolo Gussoni

Testi, Programmi, Fotografie e Disegni
Riproduzione vietata Copyright.
Qualsiasi genere di materiale inviato in Redazione, anche se non pubblicato non verrà in nessun caso restituito.

RadioELETTRONICA & COMPUTER
Rivista mensile, una copia L. 8.500, numeri arretrati lire 10.000 cadauno.
Pubblicazione mensile registrata presso il Tribunale di Monza n. 679 del 28/11/88.

Fotolito: Bassoli - Milano.
Stampa: GEMM Grafica srl, Paderno Dugnano (MI).
Diffusione: Concessionario esclusivo per l'Italia A.&G. Marco SpA, Via Fortezza 27 - 20126 Milano.
Spedizione in abb. post. gruppo III/70.

Abbonamenti: Annuale L. 54.000, estero L. 83.000.

RadioELETTRONICA & COMPUTER è titolare in esclusiva per l'Italia dei testi e dei progetti di Radio Plans e Electronique Pratique, periodici del gruppo Société Parisienne d'Édition.

Gruppo Editoriale
JCE

Gruppo Editoriale JCE srl
Sede legale, Direzione, Redazione, Amministrazione
Via Ferri 6 - 20092 Cinisello Balsamo (MI)
Tel. 02/66025.1 - Telex 352376
JCE MIL 1 - Telefax 61.27.620

Direzione Amministrativa: Walter Buzzavo

Pubblicità e Marketing
Gruppo Editoriale JCE - Divisione Pubblicità
Via Ferri 6 - 20092 Cinisello Balsamo (MI)
Tel. 02/66025.1

Direttore Pubblicità: Giuseppe Tiani

Concessionario esclusivo per Roma, Lazio e centro sud:
UNION MEDIA srl - Via C. Fracassini, 18
00198 Roma - Tel. 06/3215434 (13 linee R.A.)
Telex 630206 UNION I - Telefax 06/3215678

Abbonamenti: Le richieste di informazioni sugli abbonamenti in corso si ricevono per telefono tutti i giorni lavorativi dalle ore 9 alle 12. Tel. 02/66025311 - 66025338

Spedizioni: Daniela Radicchi

I versamenti vanno indirizzati a:
Gruppo Editoriale JCE srl, Via Ferri 6
20092 Cinisello Balsamo (MI), mediante l'emissione di assegno circolare, cartolina vaglia o utilizzando il c.c.p. n. 351205. Per i cambi di indirizzo allegare alla comunicazione l'importo di L. 3.000, anche in francobolli, e indicare insieme al nuovo anche il vecchio indirizzo.

Associato al



Consorzio
Stampa
Specializzata
Tecnica

Testata in corso di certificazione obbligatoria secondo quanto stabilito dal Regolamento del C.S.S.T.



Mensile associato
all'USPI
Unione Stampa
Periodica Italiana

Zany Golf

Questo mese iniziamo la nostra rubrica con un programma per Amiga che ha visto la luce, o meglio il drive, già da qualche mese. Si tratta di uno dei più bei programmi mai visti per Amiga, pertanto lo abbiamo studiato a fondo per presentarvi una descrizione molto dettagliata basata su una prova molto approfondita.

Il programma è Zany Golf prodotto dalla Electronic Arts. Dal nome è facile capire di che genere di gioco si tratta, tuttavia vogliamo subito dirvi che non si tratta assolutamente di una simulazione di golf, o per meglio dire è molto di più di una simulazione di golf. In realtà la struttura dei percorsi ricorda piuttosto il minigolf, ma comunque il gioco è molto più ricco di un semplice minigolf.

Lo scopo del gioco è il più ovvio: si deve

ca: se non riuscite nell'intento la partita ha termine, altrimenti passate al percorso successivo.

Ogni buca è caratterizzata dal suo par, ovvero dal numero di colpi che dovrebbero servire a un buon giocatore per raggiungere la buca. Ogni volta che raggiungete un nuovo percorso vi viene assegnato un numero di colpi pari al par della buca, più i colpi che avete avanzato dalla buca precedente.

Come potete notare si tratta di un gioco semplicissimo: non è necessario studiare pagine e pagine di manuale per imparare a giocare, tuttavia è un gioco molto sofisticato, grazie alla complessità e alla bellezza dei diversi percorsi. Per convincervi di questo potete vedere le fotografie delle diverse fasi di gioco. Comunque vogliamo descrivervi uno per uno tutti i percorsi che siamo riusciti a raggiungere.

Il primo percorso si chiama Windmill, ovvero mulino a vento. È un percorso molto semplice; quando si raggiunge una certa abilità si può scegliere la traiettoria più impegnativa che implica il passaggio tra le pale del mulino. In questo modo si vince un tiro supplementare, e se si è fortunati si porta a termine il percorso con un solo colpo.

Il secondo percorso è Hamburger. Si tratta di un percorso appetitoso poiché la buca si trova sotto a un hamburger, inoltre al primo colpo si urta una bottiglia di ketchup che schizza salsa su tutto il percorso.

La terza buca, Walls, è apparentemente facile poiché si deve semplicemente mandare la pallina contro un muro in movimento in modo che, rimbalzando su di esso, si porta in prossimità della buca. Più facile a dirsi che a farsi, poiché per ottenere il risultato desiderato si deve prendere il tempo del movimento dei diversi muri in modo da urtare quello giusto.

La quarta buca è a nostro avviso la meno divertente: si tratta di Pinball (flipper). La pallina si muove su un flipper, tuttavia i suoi movimenti sul piano inclinato sono poco accurati, e il movimento delle palette è poco preciso.

La quinta buca è molto facile, a patto di capirne il funzionamento. Si chiama



*Zany Golf.
Una bella
immagine
introduttiva.*

mandare una pallina in una buca facendole seguire un percorso ricco di ostacoli di vario genere. Il gioco è interamente comandato dal mouse, tranne che per la scelta iniziale del numero dei giocatori, che può variare da uno a quattro.

Per colpire la pallina si deve portare il cursore sulla pallina, si preme il tasto sinistro del mouse, si porta il puntatore dalla parte opposta alla direzione nella quale si desidera effettuare il colpo e infine si rilascia il tasto.

La potenza del colpo dipende da quanto vi siete allontanati dalla pallina con il cursore: più vi spostate, più la pallina viene scagliata lontano.

Avete a disposizione un determinato numero di colpi per tentare di portare a termine il percorso raggiungendo la bu-

Fans, cioè ventole. Pur essendo molto lungo il percorso tra il punto iniziale e la buca, è possibile cavarsela con un solo colpo se si è molto abili, al massimo due. In molti angoli del percorso sono disseminate delle ventole che potete azionare muovendo il mouse. Tali ventole sono disposte in modo da spingere la pallina verso la buca, pertanto imparando a dosare la velocità delle ventole è facile raggiungere il termine del percorso.

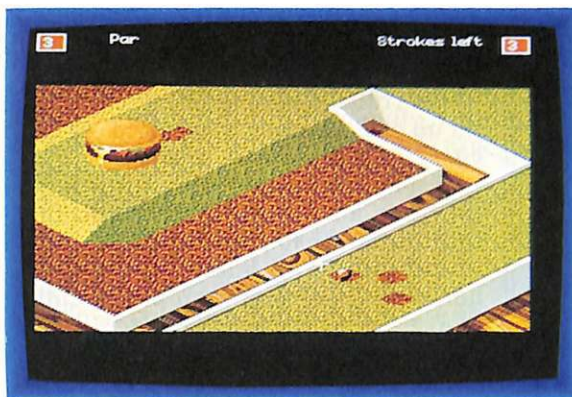
Anche nel sesto schema, Carpets, è possibile cavarsela con poco. Il percorso è infatti disseminato di tappeti (in inglese carpets) quando la pallina si trova su un tappeto basta muovere il mouse per spingerla nella direzione desiderata.

Il settimo schema è più impegnativo. Si chiama Castle, e prevede due possibili traiettorie: se siete molto abili potete cercare di entrare dal cancello del castello posto in cima ad una collinetta. Tale ingresso si chiude periodicamente, tuttavia se riuscite a penetrare da quell'apertura guadagnate un colpo. In caso contrario potete accedere da due buche. Dopo questo primo ostacolo dovete oltrepassare un piccolissimo labirinto per giungere alla buca finale.

Passiamo all'ottavo schema, l'ultimo di difficoltà accettabile. Si tratta di Anthill, la collina delle formiche, o meglio il formicaio. In questo schema la buca è posta in cima a un formicaio, e la difficoltà consiste nel dosare in modo corretto il colpo in modo da non scavalcare la sommità della collinetta.

Non è molto difficile raggiungere la buca, poiché alla base della collina lungo tutti i lati si trovano dei respingenti che potete azionare premendo il tasto del mouse, che spingono la pallina verso la collina. È solo una questione di tempismo. Dimenticavamo di dirvi che la buca è mobile, ma non si tratta di un problema.

E adesso viene il bello: il nono schema, Energy. Lo scenario è degno del migliore film di fantascienza (non vi abbiamo ancora detto che la grafica di questo gioco è eccezionale. E il sonoro ... è all'altezza della grafica!). Pare che i grafici della Electronic Arts si siano sbizzarriti per creare un eccellente panorama tecnologico ricco di computer, laser, cannoni elettronici, barriere luminose e chi più ne ha



*Zany Golf.
Lo schema
di gioco
che
stuzzica
l'appetito.*

più ne metta. Peccato che manchino le istruzioni per affrontarlo, deliberatamente omesse per rendere lo schema più avvincente. Noi le abbiamo trovate e ve le diciamo, ma invitiamo chiunque intenda davvero divertirsi con Zany Golf a non leggerle e a procedere oltre.

Lo schema è disposto su due piani: si parte dal piano inferiore e la buca si trova sul piano superiore. Per accedere al piano superiore è necessario premere due pulsanti che si trovano sul computer visibile sul lato destro dello schermo, dopodiché ci si deve portare su una piattaforma circolare dove la pallina si smaterializza per poi ricomparire su una piattaforma simile al livello superiore. Superata questa prima difficoltà si deve passare su uno scivolo per arrivare alla piattaforma dove si trova la buca. E qui viene il difficile, infatti la buca finale è circondata da buche finte che spediscono la pallina nuovamente al piano inferiore.

Il par del percorso è cinque, ma ci pare decisamente irrealistico. Noi siamo riusciti



*Zany Golf.
Attenzione
alle pale
del mulini
alla prima
bucca.*

ad arrivare a questo schema con ben 15 palline e purtroppo non siamo riusciti a portarlo a termine.

Dalla tabella segnapunti sembra che ci sia un decimo schema, tuttavia esso non viene indicato ed è un mistero.

Ci sembra di avervi detto tutto di questo eccezionale gioco: vi riassumiamo le caratteristiche fondamentali. La grafica è eccellente, con un effetto di scrolling validissimo (i percorsi sono più grandi di uno schermo). Il sonoro è pure eccellente, specialmente alla fatidica nona buca. La dinamica del gioco è ottima, originale e divertente. Vogliamo trovare a ogni costo un difetto? Volendo ben guardare la procedura di caricamento è un poco seccante, poiché al termine di ogni schema si de-

ma, dopodiché passeremo ai particolari di gioco. Con Microprose Soccer è possibile giocare secondo quattro diverse modalità. Se giocate da soli dovete scegliere la Microprose International Challenge che vi permette di sfidare una nazionale. Se invece giocate con uno o più amici avete altre tre possibilità: la partita amichevole, il torneo di coppa del mondo, che va molto di moda in questi giorni a causa dell'avvicinarsi di Italia 90, o il campionato nazionale.

Con Microprose Soccer dunque potete dare il via al primo vostro personale campionato del mondo di calcio, organizzato quasi come quelli veri. Ci saranno, infatti, i gironi eliminatori, con le squadre teste di serie e quelle cosiddette materasso (ma le sorprese non mancheranno!); si giocheranno, giornata dopo giornata, tutte le partite dei vari gironi, finché non risulterà la situazione finale e l'elenco delle rappresentative ammesse alla seconda fase; verranno disputate anche le partite successive, dopo aver sorteggiato gli accoppiamenti per i quarti di finale e per le semifinali, quindi verrà disputata la finale e si conoscerà la nazionale campione del mondo.

Se volete divertirvi a riscrivere la storia del calcio mondiale, potete selezionare l'opzione Inizio nuovo torneo, e immaginare la sede e la data che di volta in volta hanno caratterizzato le edizioni di questa grande manifestazione sportiva, la più importante nel pianeta-calcio.

Passiamo all'aspetto più tecnico del programma.

Uno degli elementi più divertenti del programma è il Control Panel, dal quale potete scegliere manualmente o automaticamente alcuni parametri, quali il tempo meteorologico e il tipo di effetto da imprimere al pallone chiamato in modo molto azzecato Banana Power, la durata della partita che può variare dai 2 ai 12 minuti, il ricorso al replay e l'accompagnamento musicale durante una partita (Off/On).

Tutti questi effetti sono davvero molto spettacolari, se si attivano gli effetti meteorologici può capitare di giocare sotto la pioggia, con addirittura i tuoni e i lampi realizzati in modo molto realistico.

Il Banana Power è essenziale per chi

Microprose Soccer: Goal sicuro il portiere è uscito, ma è stato dribblato.



ve attendere per circa mezzo minuto il caricamento del percorso successivo.

Abbiamo riportato questo particolare solo per essere più obbiettivi, tuttavia il giudizio su questo gioco non può che essere: "Non l'avete ancora comprato?".

Microprose Soccer

Se Zany Golf è uno dei migliori giochi visti ultimamente per Amiga, certamente il Soccer della Microprose lo segue a ruota. Questo gioco è disponibile sia nella versione per C64 sia per Amiga. Le due versioni sono sostanzialmente identiche: è evidente che la grafica del buon vecchio C64 non è all'altezza di Amiga, tuttavia i programmatori hanno fatto un eccellente lavoro in entrambi i casi. Vediamo dapprima qualche particolare sull'impostazione del program-

ama il tiro a effetto: è consigliabile non utilizzare la massima potenza, poiché è possibile conferire alla palla un effetto tale da farla somigliare a un boomerang; tuttavia con un po' di esperienza è possibile ottenere dei risultati decisamente eccellenti.

Il replay è un effetto del tutto influente ai fini del gioco, tuttavia è davvero spettacolare, per cui vi consigliamo di lasciarlo sempre inserito. È possibile addirittura vedere il nastro del videoregistratore durante il riavvolgimento.

Terminiamo la descrizione di Microprose Soccer illustrandovi tre movimenti essenziali di gioco: innanzitutto, per togliere la palla all'avversario, è possibile entrare in scivolata premendo il tasto Fire. Si tratta di un tackle molto bello a vedersi, ma trova un'applicazione molto più frequente che nel vero calcio (le caviglie elettroniche sono molto meno sensibili alle botte di quelle vere, provare per credere). Il tiro è scoccato con la pressione del tasto Fire: la potenza del tiro e il suo effetto (pallonetto, effetto a rientrare, eccetera) dipendono dal tempo che tenete premuto il tasto.

Anche il portiere è controllato con il tasto Fire; il computer decide il momento in cui il controllo del joystick passa dai difensori al vostro numero uno.

È necessario effettuare rinvii tesi e forti, altrimenti potreste trovarvi in difficoltà, essendo tutta la squadra sbilanciata in avanti.

Microprose Soccer è a nostro avviso la migliore simulazione di calcio esistente sia per C64 sia per Amiga. D'altra parte c'era da aspettarselo, vista l'abilità dei tecnici Microprose a creare programmi di simulazione. Certamente gli appassionati non vorranno lasciarsi sfuggire una tale realizzazione, tuttavia siamo sicuri che Microprose Soccer può essere apprezzato anche da chi non ama il calcio in quanto si tratta davvero di un ottimo videogioco.

Circus Attraction

Non possiamo sempre parlarvi di videogiocchi fantastici, quindi dopo esserci abbondantemente dilungati nella descrizione dei due migliori videogiocchi vi-

sti ultimamente passiamo ai programmi normali. Primo fra tutti, non solo per qualità ma soprattutto per originalità, è Circus Attraction, un gioco che ci introduce nell'affascinante e variopinto mondo del circo. Il giocatore è messo alla prova in tutte le discipline che da secoli tengono col fiato sospeso il pubblico di bambini e genitori di fronte allo spettacolo sotto il tendone; con questo programma acrobazie, funambolismi e gli immancabili clown possono allietarvi le serate anche a casa vostra, nel caso il circo vero non abbia raggiunto la vostra città.

Unica sostanziale differenza consiste nel fatto che in Circus Attraction non ci si limita ad essere spettatori passivi, bensì bisogna creare lo spettacolo, essendo



Microprose Soccer. Una fase culminante della partita.

coinvolti personalmente nelle tipiche attrazioni dell'arte circense.

Sarete alle prese con spericolati salti sul trampolino elastico, passeggiate mozzafiato su un filo teso a svariati metri da terra, intense esibizioni da giocoliere, rischiosissimi lanci di coltelli e fantasiose acrobazie travestiti da pagliacci.

Passiamo ai particolari prettamente tecnici dicendo che il gioco si articola in due fasi distinte. La prima parte consiste nell'addestramento alle varie discipline o, se non siete completamente digiuni in materia, in un allenamento per affinare la vostra tecnica. Quando vi sentirete pronti potrete inscenare lo spettacolo vero e proprio esibendovi consecutivamente in tutte le attrazioni. Come ogni videogame che si rispetti, anche in questo programma vi è un criterio di giudizio, in questo caso per misurare la vostra

abilità nello spettacolo: ogni prova sarà valutata in base a determinate caratteristiche. Per esempio, alla fine della prova di equilibrio sulla fune verrà attribuito un punteggio che tiene conto del cammino percorso e dei bonus aggiudicatisi.

La qualità dello spettacolo sarà espressa alla fine da una media di tutti i punteggi ottenuti in tutte le singole discipline. Affinché l'esito conclusivo non sia eccessivamente influenzato da una prova riuscita particolarmente male, il programma offre un'altra opportunità di riparare: due esibizioni a vostra scelta (oppure due volte la medesima) possono essere ripetute in modo da acquisire un punteggio migliore. Vediamo in modo più dettagliato le cinque prove che compongono lo spettacolo:

- **Salti con il trampolino elastico:** il giocatore deve compiere in un determinato tempo quante più evoluzioni possibili, selezionando con il joystick le diverse piroette o i salti mortali. Il punteggio può essere incrementato colpendo, durante i salti, dei palloncini che sovrastano il trampolino.

- **Passeggiata sul filo:** una graziosa fanciulla deve compiere un determinato tragitto fra gli estremi di una fune tesa a vari metri di altezza. I movimenti del joystick fanno sì che la ragazza mantenga l'equilibrio allargando le braccia in modo alterno. Vari oggetti sospesi sul filo permettono, se presi con un salto, di incrementare il punteggio.

- **Giocoliere:** la vostra assistente vi lancia a intervalli regolari delle palline che dovrete mantenere in aria facendole roteare fra le mani, e se necessario usando anche i piedi. Il punteggio è dato in base a quanto tempo riuscirete a resistere senza far cadere le biglie. Il joystick coordina il movimento delle mani.

- **Lancio dei coltelli:** la vostra assistente è legata a una ruota che gira vorticosamente innanzi a voi; dovrete cercare di colpire la ruota il più vicino possibile alla sagoma della ragazza. Un mirino governato dal joystick vi aiuterà nell'azione.

- **Acrobazie dei clown:** tre clown compiono evoluzioni catapultandosi da un trampolino all'altro; il punteggio si acquisisce in base al numero di passaggi e alla quantità di oggetti raccolti in aria durante i salti.

Circus Attraction è prodotto dalla Golden Goblins e può essere giocato anche da due persone contemporaneamente: in questo caso sullo schermo appariranno due artisti che non faranno due spettacoli indipendenti, bensì coordineranno le loro esibizioni per aumentare il gradimento del pubblico.

La grafica è curata in ogni particolare ed anche i movimenti degli artisti sono pressoché realistici. Anche la musica, che non poteva non essere quella tipica di ogni spettacolo circense, è particolarmente piacevole.

The running man

In Italia non è ancora uscito il film, ma è già disponibile il gioco ispirato all'ultima fatica di Arnold Schwarzenegger. Si tratta di *The running man*, una produzione Grandslam.

L'anno è il 2019 e gli Usa sono diventati uno stato totalitario. La libertà personale è stata dimenticata da tempo e la televisione è l'ultimo svago rimasto alla popolazione. La trasmissione più seguita è appunto "The running man" (uomo in fuga) un cinico gioco nel quale il protagonista deve sopravvivere agli efferati attacchi di quattro diabolici personaggi: Subzero, il campione di hockey su ghiaccio che lancia dischi esplosivi; Buzzsaw, che con la sua catena può staccare la testa in un colpo solo; Dynamo, il misterioso uomo-albero; infine Fireball, capace di arrostiti qualsiasi cosa gli si ponga innanzi con una semplice fiammata.

Lo scopo del gioco è arrivare indenni in fondo al terribile percorso fra i sobborghi e il centro di una Los Angeles ormai imbarbarita; superata l'ultima prova entrerete negli studi televisivi per raggiungere Killian, il sadico ideatore del gioco, e guadagnarvi così la libertà.

Durante il cammino incapperete in armi e oggetti vari che vi saranno utilissi-

mi per far fronte agli attacchi. Se ciò non accade dovrete far affidamento solo sulla vostra abilità nell'usare le mani; durante i corpo a corpo due barre luminose esprimono lo stato di salute dei due contendenti. Per risanare le ferite provocate dai colpi subiti ci sono due modi: il primo consiste nell'impossessarsi delle cassette di medicinali disseminate sul cammino, nel secondo si tratta di risolvere in un tempo dato una prova di coordinazione in cui bisogna accoppiare ben otto simboli.

Il punteggio finale si basa sulla quantità di denaro puntata su di voi dai telespettatori: se combatterete bene e supererete facilmente gli avversari il pubblico scommetterà maggiormente.

The Running Man è un gioco avvincente, adatto soprattutto agli appassionati di giochi d'azione. Il soggetto non è particolarmente innovativo, ma il programma riesce ugualmente a coinvolgere in modo notevole il giocatore. La grafica è apprezzabile, come del resto la varietà di movimenti di cui il personaggio che vi rappresenta sullo schermo è capace. Molto probabilmente è più interessante il gioco del film!

Silkworm

Silkworm è il classico shoot-em-up game, locuzione inglese che identifica quei giochi in cui l'unica azione è quella di sparare all'impazzata, per colpire qualsiasi cosa si aggiri nei paraggi.

Eccone la trama: il governo sulla terra è retto da una pacifica alleanza dei cinque continenti. E in atto un tentativo di colpo di stato da parte di un gruppo di generali, che rischia di riportare la terra ai tempi in cui i rapporti fra gli stati erano determinati dai precari equilibri creati dagli arsenali nucleari. Per sventare il piano l'Alleanza ha scelto il miglior pilota di elicotteri e di jeep da combattimento, e lo ha dotato dei più potenti mezzi d'attacco in circolazione. Al comando di essi, il prescelto si addenterà nelle roccaforti nemiche fino a scovare uno ad uno i generali rivoltosi, in modo da eliminarli.

Prima dell'inizio dell'azione un menù, permetterà di scegliere se avventurarsi

con l'elicottero o con la jeep; quest'ultima offre la possibilità di indirizzare il fuoco in qualsiasi direzione, per ovviare all'handicap di potersi muovere unicamente sulla superficie terrestre. Inoltre è in grado di scavalcare con un balzo eventuali ostacoli che non siano stati eliminati con il cannone di bordo.

L'elicottero è molto maneggevole ma può sparare in due sole direzioni, in avanti e in basso. Il terreno è cosparso di cannoni nemici e di mine che, una volta esplose, lasciano una nuvola di gas; passandoci dentro si ottiene per un breve periodo di tempo la totale protezione da qualsiasi proiettile nemico. Immergendosi in due nuvole consecutive si dispone automaticamente di una smart bomb ca-



Circus Attraction. Equilibrio e freddezza per passare da un capo all'altro della fune

pace di annientare tutto ciò che si trova sullo schermo. Ogni livello di gioco termina con l'annientamento di un generale che appare sul suo veicolo corazzato.

Per concludere un livello si ha a disposizione un determinato tempo: se si è particolarmente veloci i secondi residui si tramutano in bonus per incrementare il punteggio finale.

Silkworm, creato dalla Virgin Mastertronic, possiede una giocabilità non elevatissima a causa dell'estrema ripetitività dell'azione; né l'opportunità di scelta fra due modalità di gioco, né l'eccellente grafica possono fare molto per far passare in secondo piano questo aspetto negativo di un programma che, tuttavia, offre buoni spunti per divertirsi, anche se non a lungo.

**Gianni Arioli
Massimiliano Del Rio**

**C64E**

ACCESSORI

La vita degli appassionati e dei professionisti in Computer Art diviene di giorno in giorno più facile grazie alle rivoluzionarie innovazioni tecniche. Questo mese vi presentiamo un sofisticatissimo accessorio per disegnare su Amiga, un utile strumento per fare facilmente grafica anche con il Commodore 64 e una interessante rassegna di accessori vari

Tavola, penna e calamaio

La grafica di Amiga eccita la fantasia dei progettisti hardware che immettono sul mercato ogni mese nuove idee. Questo mese iniziamo la nostra prova di accessori con uno strumento utile.

EasyL.

EasyL è una tavoletta grafica per Amiga e per IBM compatibili. EasyL non è una normale tavoletta grafica, bensì un'eccezionale tavoletta grafica, e mai nome è

stato più appropriato al prodotto. Questo accessorio, infatti, rende davvero la vita facile (easy) a qualsiasi disegnatore. Probabilmente il prezzo limiterà l'uso di EasyL ai soli professionisti di com-





C64E

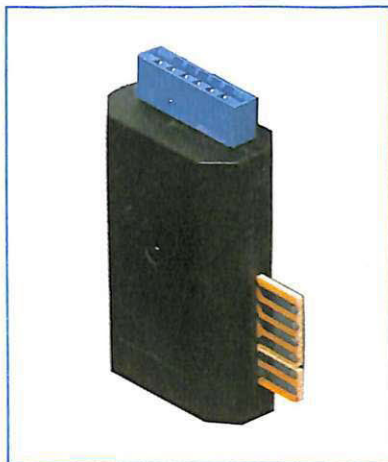
puter graphic, tuttavia siamo certi che anche chi non si occupa strettamente di disegno sarà fortemente tentato all'acquisto di questa eccezionale periferica.

Siamo altrettanto certi che gli amanti del disegno elettronico non potranno sicuramente fare a meno di acquistare Easyl, poiché nessun altro accessorio può completare in modo efficace le ottime qualità grafiche di Amiga. Easyl è un prodotto eccezionale anche per quanto riguarda il luogo di provenienza, il Canada. Easyl è prodotta dalla Anakin Research Inc., che ha sede a Toronto.

La tavoletta vera e propria è una struttura di plastica molto elegante, realizzata in modo da accompagnare il design di Amiga. Easyl è un parallelepipedo di plastica piuttosto sottile dotato di due pulsanti che emulano i tasti del mouse. La parte sensibile occupa la quasi totalità della superficie maggiore. Nonostante l'aspetto sottile Easyl è piuttosto massiccia e fornisce una netta impressione di solidità.

Easyl è una tavoletta grafica presso-sensitiva in formato A4. Questo significa che potete appoggiare un foglio di formato A4 sulla superficie di lavoro della tavoletta, e iniziare a disegnare con una semplice penna o matita. Easyl sentirà dove state passando la penna e trasferirà le informazioni al computer. La risoluzione è molto buona: 1024 x 1024 pixel. Poiché la massima risoluzione dello schermo di Amiga ammonta a 640 x 512 pixel (Easyl non può lavorare con l'OverScan) è evidente che a ogni pixel dello schermo corrispondono circa due pixel della tavoletta. Il software di gestione, inoltre possiede una routine di filtro che permette di trasformare nel modo più efficace la lettura della tavoletta in coordinate dello schermo.

Procediamo con una descrizione dettagliata della prova effet-



tuata con Easyl. Easyl è fornita in quattro differenti versioni, tre per Amiga (500-1000-2000) e una per Ibm. È stato necessario produrre queste differenti versioni poiché l'interfaccia di Easyl deve essere collegata direttamente al bus dati del computer, e quindi nel caso di Amiga 2000 o Ibm la scheda può essere inserita direttamente nel computer, mentre per gli altri modelli di Amiga l'interfaccia deve essere collegata esternamente. Noi abbiamo provato la versione per Amiga 2000. La confezione (molto grande e accuratamente imbottita di gommapiuma) comprende la tavoletta, la scheda di interfaccia, il manuale originale in inglese, la traduzione in italiano del manuale e un dischetto con il software.

L'installazione è molto semplice: si deve aprire Amiga svitando con pazienza tutte le viti che trattengono il coperchio, si inserisce la scheda in un qualsiasi slot libero avendo cura di togliere lo sportellino per la porta esterna, si richiude Amiga. La scheda è molto complessa e lascia intendere che siamo davanti a uno strumento molto sofisticato.

Effettuato il montaggio, sul retro di Amiga compare una porta Din a nove poli nella quale dobbiamo inserire lo spinotto di cui è dotata la scheda, e il gioco è fatto.

Non ci stancheremo mai di ripetere che un ottimo hardware è totalmente inutile se non è dotato di un valido software: Easyl risponde a queste caratteristiche in modo molto soddisfacente.

Il software in dotazione può essere suddiviso in due parti indipendenti che abbiamo provato separatamente.

La prima parte è un editor grafico dedicato alla tavoletta, e sicuramente si tratta del software che permette di ottenere i migliori risultati. Non vogliamo entrare nei dettagli del programma di gestione: Easyl (anche il programma si chiama così) è un classico editor grafico dotato di tutto quanto può servire; si tratta di un programma meno potente di De Luxe Paint II, tuttavia dispone di tutti i comandi utili per il disegno, richiamabili naturalmente dai menù a discesa. Il programma è disponibile in due versioni, Ntsc e Pal. La prima versione è da utilizzare sui monitor americani che non possono superare la risoluzione di 400 linee in quanto è imposta dallo standard Ntsc, mentre la seconda consente agli utilizzatori europei di raggiungere le 512 linee. Il programma può girare anche su Amiga 500-1000 dotata di soli 512 Kb di memoria, tuttavia alcune istruzioni e soprattutto la massima risoluzione (640 x 512) non sono in grado di funzionare correttamente.

Disponendo di soli 512 Kb può essere utile utilizzare la versione Ntsc che utilizza una quantità inferiore di Ram.

La caratteristica migliore del programma consiste ovviamente nel fatto che è possibile gestire in modo molto dettagliato le possibilità offerte da una tavoletta. Innanzitutto Easyl lavora con due puntatori, uno è il solito puntatore gestito da Intuition e controllato dal mouse che serve per controllare i menù a discesa, mentre l'altro è controllato dalla tavolet-



C64E

ACCESSORI

ta e serve per disegnare.

È possibile disegnare in modo Portrait (ritratto) oppure Landscape (panorama).

Nel primo caso la tavoletta è posta verticalmente, ovvero nella posizione adatta per l'esecuzione di ritratti, mentre nell'altro caso la tavoletta è utilizzata orizzontalmente. In linea di massima è più comodo il modo Landscape poiché si adatta meglio alle proporzioni dello schermo, tuttavia la possibilità di scelta è sempre molto gradita.

I menù funzionano nel consueto modo molto intuitivo del tutto familiare agli utenti di Amiga. Vediamo invece, come si procede per disegnare. Non potrebbe essere più facile: si dispone un foglio sulla superficie sensibile della tavoletta, si prende una penna e si disegna. Si deve avere cura di non spostare la penna in modo troppo veloce altrimenti Easyl non è in grado di seguirla, tuttavia anche se superate la velocità di tracciamento del programma, Easyl è in grado di ricordare dove siete passati, in quanto dispone di un buffer. La velocità di lettura dichiarata ammonta a 250

punti al secondo, ovvero un valore di tutto rispetto, tuttavia si deve ricordare che la lettura della tavoletta deve essere elaborata prima di essere convertita in pixel. In pratica è possibile disegnare in modo abbastanza naturale, basta avere cura di evitare rapidi colpi di penna.

La precisione è davvero eccezionale, per testarla abbiamo provato a scrivere qualcosa sulla tavoletta e abbiamo ritrovato sullo schermo la scritta con la nostra calligrafia riprodotta in modo molto fedele. Per effettuare un ulteriore test abbiamo provato a ricalcare qualche disegno preso da una rivista: siamo davvero rimasti a bocca aperta davanti al risultato ottenuto.

Il vantaggio più importante offerto dall'uso del programma dedicato consiste però nella possibilità di servirsi dei comandi diretti. Un lato della superficie sensibile della tavoletta è occupato da una serie di caselle disposte su due righe raffiguranti i 32 colori di default.

Naturalmente è possibile disegnare a 32 colori solo se la risoluzione orizzontale ammonta a 320 pixel, altrimenti si possono utilizzare solo 16 colori.

Per selezionare un colore è sufficiente premere con la penna sulla casella raffigurante la sfumatura desiderata. Se si disegna facendo uso di un gran numero di colori questo tipo di comandi diretti si rivela davvero molto utile e consente di risparmiare moltissimo tempo. Naturalmente è possibile modificare i colori di default utilizzando il comando Colours dal menù Edit, e impostando a proprio piacimento l'intensità di rosso, verde e blu (RGB=Red Green Blue) di ciascuno dei 32 colori.

È ovvio che Easyl prevede numerosissimi altri comandi, tuttavia abbiamo voluto indicarvi solo i comandi strettamente le-



gati all'uso della tavoletta ricordandovi che sono comunque disponibili tutti i comandi più importanti per un editor grafico.

Vediamo invece come sia possibile utilizzare Easyl con un qualsiasi altro programma anche di tipo non grafico. Sul disco di sistema esistono alcune versioni di un programma chiamato con il nome generico di Driver.

Il Driver serve per utilizzare Easyl come un mouse, ovvero a controllare direttamente il puntatore del mouse.

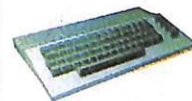
I driver sono compatibili con qualsiasi altro programma scritto per Amiga Dos 1.2, infatti non fanno altro che assegnare il controllo del puntatore alla tavoletta grafica. Questo comunque non significa che il mouse sia disabilitato: dopo avere attivato un Driver è possibile continuare a utilizzare anche il mouse.

È ovvio che se si tenta di utilizzare contemporaneamente il mouse e la tavoletta non si ottiene nulla di utile in quanto le due periferiche si disputano il controllo del puntatore.

Abbiamo parlato di differenti versioni di Driver, poiché i tecnici della Anakin sono stati tanto saggi da prevedere un utilizzo differenziato della tavoletta da parte dei mancini e dei destrorsi.

Il Driver dunque è disponibile in una versione per destrorso che prevede l'uso della tavoletta con i pulsanti disposti sul lato sinistro,





C64

e una versione per mancini che consente di utilizzare la tavoletta in senso inverso.

Oltre a tutti questi driver standard sono disponibili, sempre in due versioni per destrorsi e mancini, altri tipi di driver che consentono un uso leggermente differente della tavoletta che può rivelarsi utile a seconda del tipo di programma che intendete utilizzare, per esempio con il RRightdriver e RLeftdriver, è possibile evitare la pressione del tasto per disegnare.

Abbiamo provato a utilizzare i Driver con De Luxe Paint e con Photon Paint, e anche in questo caso i risultati sono stati eccellenti. L'unico problema è stato posto da Photon Paint in quanto il programma (la cui caratteristica più rilevante è la gestione del modo Ham) è piuttosto lento, e fa piuttosto fatica a tenere il passo della tavoletta. Si tratta in realtà di una debolezza del programma, e non della tavoletta, e comunque è un dettaglio poco rilevante che non toglie nulla alle possibilità offerte dalla tavoletta. Dobbiamo davvero concludere la prova di Easy1 in modo scontatissimo: sia hardware sia software sono davvero eccellenti e costituiscono uno strumento assolutamente fondamentale per chiunque desidera utilizzare Amiga in modo grafico creativo. La precisione con la quale è possibile disegnare, unita alle ottime possibilità di controllo fanno di Easy1 una periferica di gran lusso. Di gran lusso è anche il prezzo, purtroppo, e siamo sicuri che solo questo particolare può frenare la diffusione di Easy1. Per ulteriori informazioni rivolgersi alla Flopperia a Milano in via Montenero 31, tel. 02/55180484

Penna ottica

Abbandoniamo il fantastico mondo di Amiga, dove il compu-



ter graphics diventa realtà, per vedere qualche accessorio per Commodore 64.

Questo mese facciamo una panoramica su tanti piccoli accessori che contribuiscono attivamente a dare al C64 l'immenso successo che si è meritato nella sua lunga carriera (pochi anni in realtà, ma quale computer ha riscosso un simile successo così a lungo?). Restiamo in tema grafico, dopo la tavoletta grafica per Amiga, la penna ottica per C64 prodotta dalla GP Elettronica.

Si tratta di uno strumento molto semplice, una penna con un sensore ottico in cima e un cavetto per collegarla alla porta del joystick del C64, una cassetta con il software e un foglietto di istruzioni. Il software altro non è che il famoso Blazing Paddles settato in modo da gestire la penna ottica. Per chi non lo conoscesse diremo rapidamente che si tratta di un buon editor grafico, di valore simile al più famoso Koala Painter. Con Blazing Paddles è possibile disegnare a mano libera, tracciare linee, cerchi, ovali e rettangoli. Non mancano le opzioni di Fill, Spray e tutti quei comandi che lo rendono un programma

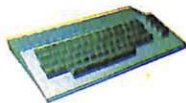
particolarmente efficace.

La penna ottica è in grado di controllare il puntatore di Blazing Paddles, il che significa sia utilizzare i vari comandi, sia disegnare. La caratteristica più importante di una penna ottica è la precisione.

La penna ottica è in grado di svolgere le sue funzioni sfruttando il fatto che il computer, o meglio il chip che sovrintende la grafica del C64, conosce la posizione del pennello elettronico, che crea l'immagine sullo schermo.

La penna ottica manda un segnale al computer, quando vede passare il pennello elettronico, e il computer è in grado di ricavare la posizione della penna semplicemente leggendo la posizione del pennello dal chip grafico.

Poiché il pennello elettronico si muove con una frequenza verticale di 50 Hz e con una frequenza orizzontale di circa 16000 Hz, è ovvio che il computer non ha problemi a rilevare la posizione verticale della penna, mentre la posizione orizzontale è piuttosto critica. La penna ottica riflette molto bene questa caratteristica tipica di qualsiasi penna ottica, infatti la lettura verticale è mol-

**C64E**

ACCESSORI

to precisa, mentre la lettura orizzontale è un po' tremolante.

Questo significa che la penna ottica funziona molto bene quando viene usata per puntare delle icone, come per impartire i comandi di Blazing Paddles, mentre a nostro avviso non è sufficientemente precisa per consentire buoni risultati in fase di disegno. D'altra parte si tratta di una periferica decisamente economica: la Penna Ottica Grafica è in vendita a sole 25mila lire.

Altri accessori per C64

Abbandoniamo ora la grafica per dedicarci alla prova di tanti piccoli accessori per C64.

• Allinea testine

Chi possiede un C64 dotato di registratore sa quanto possa diventare fastidioso caricare un programma se il registratore non è in perfette condizioni.

È facile per esempio iniziare a caricare un programma e vedere interrotte la procedura di caricamento solo dopo alcuni minuti, con il risultato che si deve ricominciare da capo.

I motivi per cui un registratore non funziona correttamente sono essenzialmente tre: può avere la testina sporca, magnetizzata, oppure disallineata.

Come si vede, la testina è quasi sempre la responsabile del cattivo funzionamento di un regi-

stratore. Se è sporca, è molto facile pulirla, servendosi di uno straccetto morbido e pulito imbevuto di alcool.

Se è magnetizzata, il che accade dopo molte ore di uso del registratore, specialmente se usate nastri di cattiva qualità, è necessario servirsi di un demagnetizzatore.

Il difetto più grave comunque resta il disallineamento, si tratta anche del difetto più subdolo, poiché capita di poter leggere facilmente i file scritti con lo stesso registratore, mentre è impossibile leggere i file scritti con un altro registratore, o i nastri in commercio.

Se si procede troppo tardi al riallineamento c'è il rischio di non essere più in grado di leggere i propri programmi scritti con la testina disallineata. È dunque importante avere sempre il registratore a posto, e a questo scopo ci si può servire del Super Allineatore Elettronico, anch'esso prodotto dalla GP Elettronica. La confezione comprende un piccolo accessorio da inserire tra il registratore e il computer una cassetta e un cacciavite da utilizzare per ripristinare il corretto azimuth della testina, oltre naturalmente alle istruzioni per l'uso. Allineare le testine con il dispositivo in questione è un'operazione molto semplice: si inserisce l'accessorio di controllo tra il registratore e il computer, si mette la cassetta in dotazione nel registratore e si preme il tasto Play. Se la testina del registratore è ben allineata il led presente sull'allineatore si accende. Se non si accende oppure lampeggia, si deve agire con il cacciavite sull'apposita vite del registratore finché non si ottiene una perfetta accensione.

Un'operazione davvero semplice e molto utile. In fotografia potete ammirare altri prodotti GP Elettronica.



• Il prova joystick

Il prova joystick serve per controllare l'efficienza di un joystick, funziona con una pila da nove volt, e dispone di cinque led tramite i quali è possibile sincerarsi del funzionamento dei cinque contatti del joystick.

Ci sembra un accessorio davvero superfluo e inutile, ma qualche amante dell'ordine e della precisione potrebbe preferire usare il prova joystick anziché un qualsiasi videogioco.

• Joystick

È molto bello invece il joystick trasparente: si tratta di un joystick normalissimo a contatti, tuttavia è realizzato in plastica trasparente che gli fornisce un aspetto futuristico.

Per chi ama il classico è comunque disponibile lo stesso modello di joystick realizzato in plastica nera.

• Paddles

Terminiamo con le paddle anch'esse di buona fattura, dotate di un simpatico astuccio, utili in giochi come il mitico Arkanoid e cloni vari.

Nella vasta gamma di prodotti della GP Elettronica troviamo cavi di prolunga per joystick e sdoppiatori.

Gianni Arioli



ELECTRONICS PERFORMANCE

• PERMUTE
• RIPARAZIONI
• ASSISTENZA

OFFERTE SPECIALI • VENDITA PER CORRISPONDENZA • SCONTI RISERVATI AI RIVENDITORI

ARTICOLO	PREZZO	ARTICOLO	PREZZO
Amiga 500 con mouse e 3 dischi	739.000 *	Joystick Flashfire con 3 spari manuali	10.000
Commodore 64 New + Registratore + Penna Ottica	339.000 *	Joystick Flashfire con 3 spari man + autofire	15.000
Commodore 128 + Registratore + Penna Ottica	350.000 *	Joystick Flashfire con autofire trasparente	19.000
Disk Drive compatibile x 64/128k	250.000 *	Joystick Flashfire con Microswitches B.	29.000
Disk Drive 1541 II	320.000 *	Joystick Albatros microswitches	49.000
Drive esterno x Amiga 500/2000 multi/disconn.	195.000 *	Joystick Flashfire x C16	15.000
Stampante Riteman x C 64/128K	390.000 *	Joyboard	49.000
Stampante per Commodore 64/128k	325.000 *	Joystick x Amstrad con sdoppiatore D.	24.000
Stampante Commodore 1230 x C64 Amiga e PC	490.000 *	Joystick x Amstrad con sdoppiatore D Microsw	32.000
Stampante Epson x Amiga e Pc	395.000 *	Joystick Digitale a sfioramento	45.000
Stampante Star Lc 10 Colore x Amiga e Pc	495.000 *	Paddles Controllers (coppia)	18.000
Monitor Monocromatico con audio (universale)	165.000 *	Adattatore x Joystick C16	5.000
Monitor colore Commodore 1802 x C64 e Amiga/m	330.000 *	Mouse x C64 128K	60.000
Monitor colore 1084 C x C64 Amiga, PC	520.000 *	Mouse x Amiga 500	95.000
Monitor Philips x C64 Amiga PC	460.000 *	Mouse Elettronico X C64 128K Amiga	45.000
TV Monitor Universale	580.000 *	Penna ottica x C64/128k cassetta o disco	35.000
Registratore Compat. x C64/128k	50.000 *	Geos x C64/128K	49.000
Registratore Commodore x C64/128k	68.000 *	Regolatore Elettronico di testine reg.	19.000
Registratore 1531 x C16	68.000 *	Fast Disk (velocizzatore + copiatore C64)	45.000
Adattatore Registratore da C16 e C64	25.000	Final Turbo IV New	79.000
Adattatore Duplicatore x registratore C16	18.000	Power Cartridge	120.000
Duplicatore x Registratore C64	18.000	Isepic Tape (sprotettore e + copiatore C64)	65.000
Modulatore x Amiga 500	45.000 *	Reflex Backup C64	49.000
Espansione di Memoria 512K Amiga 500	290.000 *	Turbo Dos	90.000
Videodigitalizzatore Amiga 500	200.000	Speed Dos 1541 II	90.000
Videon (digitalizzatore Amiga)	390.000	Digitalizzatore x C64	50.000
Mini Gen Lock (miscelatore di immagini Amiga)	290.000	CartucciaCpm x C64	100.000
Interfaccia Musicale "Midi" x Amiga 500	120.000	Cartuccia Espansione C64	150.000
Copricomputer Plexiglas Amiga 500	19.500	Rom x Stampante Mps 803	48.000
Copricomputer Plexiglas 128K	18.000	Portacassette "Posso" 15 Pz.	15.000
Copricomputer Plexiglas C64 New	16.000	Portacassette Video "Posso"	28.000
Copricomputer Plexiglas C64 C16 Vic 20	14.000	Portadischetti 5" 1/4 "Posso"	38.000
Modulatore x Vic 20	35.000 *	Portadischetti 3" 1/2 "Posso"	38.000
Cavo Start x Amiga 500 Tv e Monitor	27.000 *	Portadischetti 3" 1/2 (cont 10 Pz.)	4.500
Cavo Start C64/128k/C16/Vic 20	19.000 *	Portadischetti 3" 1/2 (cont 25 Pz.)	18.000
Cavo Monitor C64/128K/C16	25.000	Portadischetti 3" 1/2 (cont 40 Pz.)	24.000
Cavo Monitor per Amiga 500	40.000	Portadischetti 3" 1/2 (cont 80 Pz.)	28.000
Cavo Monitor Colore x PC	59.000	Portadischetti 5" 1/4 (cont 10 Pz.)	4.500
Cavo Monitor videocomposto x PC	12.000	Portadischetti 5" 1/4 (cont 50 Pz.)	24.000
Cavo Monitor videocomposto x C64	14.000	Portadischetti 5" 1/4 (cont 100 Pz.)	28.000
Cavo TV/Computer mt. 1,5	6.500	Portacassette a valigetta 16 Posti	14.000
Cavo TV/Computer mt. 3	12.000	Portacassette a valigetta 36 Posti	18.000
Filtro antidisturbo x Computer	19.000	Portacassette Componibili (multibox)	3.500
Cavo 40/80 colonne x 128k anche Skart	25.000	Dischetti 3" 1/2 Df Dd Conf. 50 Pz. (1 Mega)	CAD 2.000
Cavo Seriale x Drive e Stampante	16.000	Dischetti 3" 1/2 Df Dd Conf. 20 Pz. (1 Mega)	CAD 2.500
Cavo Centronics	35.000	Dischetti 3" 1/2 Df Dd Sciolti (1 Mega)	CAD 3.000
Alimentatore Universale	28.000	Dischetti 5" 1/4 Df Dd Conf. 10 Pz.	CAD 1.000
Caricatore Batterie e Alimentatore	39.000	Dischetti 5" 1/4 Df Dd xc PC	1.500
Alimentatore x Atari 2600	35.000 *	Dischetti 5" 1/4 Df Dd 3M	6.000
Alimentatore x Commodore C16	35.000 *	Tagliadischetti in Acciaio	16.000
Alimentatore Commodore 64	45.000 *	Tagliadischetti in Plastica	12.000
Cassette Vergini x Computer da C 10 e C 120	da 1.000	Kit Puliscitistine x Drive	15.000
Videoregistratori e cassette video	Telefonare	Giochi Disco x C 64/128/Amiga/PC/Msx ecc. da L.	10.000
Kit Pulisci testine x Videoregistratore	19.000	Giochi Cassette x C64/MSX/C16/Atari ecc. da L.	9.000

* I PREZZI INDICATI CON ASTERISCO SI INTENDONO + IVA 19%

ELECTRONICS PERFORMANCE

Via S. Fruttuoso, 16/A - Monza (S. Fruttuoso) - Tel 039/744164



In questa puntata, articolata in sei paragrafi, vedremo alcune regole sintattiche che caratterizzano il linguaggio Logo e la tradizione di parole e liste

Logo e grammatica

CAPITOLO DODICESIMO

Rispondi o non rispondi?

Quelli di voi che odiano la matematica, avranno già meditato di saltare il capitolo.

Comunque, non parleremo molto di matematica; cercheremo soprattutto di illustrare alcuni concetti, i quali diventeranno piuttosto importanti quando lavoreremo a progetti non esclusivamente grafici, cioè quando avremo a che fare con parole e liste che studieremo in seguito nel prossimo capitolo.

Un piccolissimo chiarimento a proposito dei termini *Primitiva* e *Procedura*:

- le primitive sono, come sappiamo, tutte quelle parole che il Logo sa già quando accendiamo il computer; esse svolgono ciascuna un certo compito, che non si può modificare;

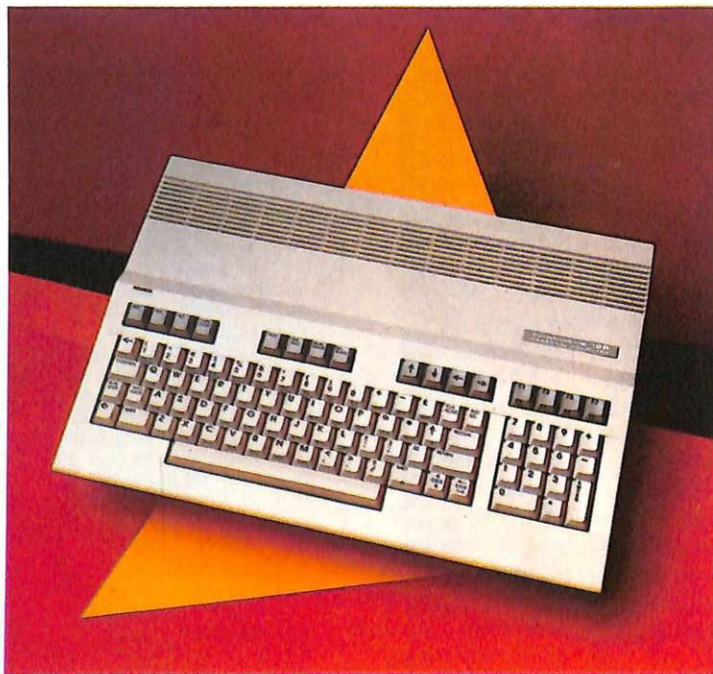
- le procedure, strettamente parlando, sono solamente quelle che definite voi in Logo, servendovi di per, e dell'editore.

Tuttavia in questo capitolo chiameremo a volte procedure l'insieme delle primitive e delle

quelle che riportano un risultato, e quelle che non lo fanno.

Tipici esempi di primitive che riportano un risultato (il cui nome tecnico è *Funzioni*) sono le quattro operazioni di addizione, sottrazione, moltiplicazione e divisione (+, -, x, /) che sono in fondo primitive come le altre, anche se vengono scritte tra i loro argomenti e non davanti.

Anche *ACaso* è una funzione, come in generale tutte le procedure che servono ad eseguire calcoli oppure operazioni su parole e liste. Tra le primitive che non riportano un risultato ci sono *Avanti*, *Destra* e in generale quasi tutte quelle che riguardano la grafica, ma anche altre molto usate come *Assegna*, *Conserva*, eccetera. Il loro compito è modificare lo stato attuale del sistema; per esempio, spostare la tartaruga o modificando il valore delle variabili. Questi vengono di solito chiamati *Effetti Collaterali* della procedura. Per esempio, la



procedure definite da voi.

Ora, si possono facilmente dividere le primitive (e le procedure) Logo in due grandi classi:

ga o modificando il valore delle variabili. Questi vengono di solito chiamati *Effetti Collaterali* della procedura. Per esempio, la



scacchiera disegnata dalla procedura Scacchiera è un suo effetto collaterale, mentre la procedura stessa non emette alcun risultato. Lanciare (ovvero chiamare) una funzione, è un po' come fare una domanda attendendosi una risposta, mentre lanciare una procedura dell'altro tipo è come dare un ordine aspettandosi molto semplicemente che esso venga eseguito.

Alcuni linguaggi di programmazione contengono esplicitamente questa distinzione tra funzioni e procedure senza risultato (ma con effetti collaterali). In Logo possiamo verificare facilmente con che tipo abbiamo a che fare semplicemente chiamando la procedura e controllando se appare un messaggio come:

RISULTATO: qualsiasi cosa.

In tal caso si tratta di una funzione, e l'uso che se ne fa normalmente è passare il suo valore (il suo risultato) come argomento, a una qualche altra procedura.

Se per caso chiediamo al Logo di calcolare qualche cosa che poi ci dimentichiamo di usare (a meno che non si tratti di un singolo comando diretto, cioè fuori da una procedura e senza niente dopo), ci chiederà:

DIMMI COSA FARE DI
XXYXXX

Ora provate con:

3 * 4 AVANTI 50
RETTANGOLO 10 20 30 40
ACASO 10 ACASO 20

e riflettete sui messaggi di aiuto.

Se invece trattate da funzione una procedura che non riporta risultato, il Logo vi comunicherà che:

YYXYXX NON RIPORTA
NULLA

Esercizi - Capitolo dodicesimo

1) Modificate la procedura 18 per cento in modo che accetti due argomenti, uno dei quali sarà la percentuale richiesta. Sulla base di questa, createne una che calcoli la spesa totale (importo + Iva), e inseritela in un'altra procedura, che vi chieda due (o più) importi e restituisca la somma dei risultati.

2) Se i numeri vi affascinano, scrivete procedure che calcolino il reciproco, la sola parte decimale (esiste la primitiva Intero) e qualsiasi altra cosa vi venga in mente. Come realizzereste una procedura per calcolare l'ennesima potenza di un numero?

3) Escogitate qualcosa per impedire a Potenza di entrare in un ciclo infinito quando le passate un argomento improprio (decimale o negativo).

4) Scrivete una procedura ricorsiva che calcoli il fattoriale di un numero. Il fattoriale è una funzione (il suo simbolo ufficiale è il punto esclamativo [!]) che, dato un numero, calcola il prodotto di quel numero con tutti quelli che lo precedono: così il fattoriale di tre è dato da $1 \times 2 \times 3$, e il fattoriale di cinque da $1 \times 2 \times 3 \times 4 \times 5$. Il fattoriale di zero, invece, viene stabilito uguale a uno, più che altro per convenzione.

5) Scrivete una procedura che riconosca i numeri divisibili per tre. Volendo, è possibile generalizzare e programmarne una che verifichi se un numero è divisibile per un certo altro numero dato. Se vi sentite dei leoni, cercate di arrivare adesso ad una procedura che riconosca i numeri primi. È tutto.

Potete verificare questo con comandi tipo:

DESTRA AVANTI 100
STAMPA 3 * ASSEGNA "X 666

In entrambi i casi la sedicente funzione (Avanti nel primo caso, e Assegna nel secondo) viene eseguita fino in fondo, ed è soltanto dopo di ciò che cominciano i guai, dal momento che il risultato atteso non compare.

Il messaggio di errore viene infatti emesso soltanto dopo l'esecuzione di Avanti o Assegna, tanto è vero che la tartaruga si sarà mossa e :X varrà 666.

Per riassumere, se guardiamo il lavoro di una procedura dall'esterno possiamo sempre descriverne l'intero comportamento nei termini di questi tre elementi:

- **Input:** letteralmente significa mettere dentro; sono i dati che passiamo alla procedura e su cui essa dovrà lavorare, in altre parole gli argomenti.

- **Output:** (mettere fuori) è quello che esce dalla procedura, e in un certo senso viene sostituito ad essa il suo valore, ovvero il Risultato. Come abbiamo visto, può benissimo mancare.

- **Effetti collaterali:** tutto il resto, tutto ciò che la procedura fa senza avere bisogno di dircelo, sempre che provochi qualche modifica nel sistema.

Funzioni aritmetiche

Il Logo Commodore dispone della capacità di trattare sia numeri interi che numeri reali (i nu-



meri reali sono quelli con il punto decimale), e possiede un certo numero di primitive per compiere operazioni sui numeri. Per un elenco completo, consultate l'appendice del Manuale Logo I.

N.B.: in italiano solitamente separiamo con la virgola la parte intera di un numero dalla sua parte decimale. In Logo invece, seguendo la tradizione anglosassone, si dovrà usare il punto.

Che cosa succede se abbiamo bisogno di eseguire un calcolo ma il Logo non ci fornisce la primitiva appropriata?

La cosa migliore è mettersi a programmare una procedura che riporti il Risultato del calcolo. Per dire ad una procedura di fare questo (cioè di passare qualcosa all'esterno) si usa la primitiva Riporta (o l'abbreviazione RI), seguita da quel che bisogna riportare; due esempi:

- una procedura che riporta il doppio di un numero:

```
PER DOPPIO :N  
RIPORTA 2 * :N  
FINE
```

- e una che, dato un numero, riporta il diciannove per cento (utile per il calcolo dell'Iva).
PER DICIANNOVE% :IMPORTO
RIPORTA (:IMPORTO * 19) / 100
FINE

Confrontate attentamente quest'ultima procedura con la seguente, che stampa l'importo anziché riportarlo, e meditate attentamente sulla differenza tra le due:

```
PER STAMPA18% :IMPORTO  
ST ( :IMPORTO * 18 ) / 100  
FINE
```

Una caratteristica importante di Riporta è che arresta comunque l'esecuzione della procedura (al pari di Stop), non ap-

pena ha calcolato il suo argomento. Quindi, per esempio, non potete aggiungere niente dopo l'unica linea del 19 per cento: la vostra aggiunta non sarebbe comunque raggiunta.

Vediamo ora un esempio con un'istruzione condizionale all'interno: una procedura che calcola il valore assoluto di un numero (cioè il valore senza segno):

```
PER VAL.ASS :N  
SE :N > 0 RIPORTA :N  
RIPORTA - :N  
FINE
```

Un po' di ricorsività

Se vi siete soffermati un attimo sull'ultima domanda dell'esercizio 2 di questo capitolo, vi sarete resi conto che il calcolo della potenza di un numero comporta uno sforzo di tipo diverso da quello, per esempio, della risoluzione di una percentuale.

Là avevamo un numero prefissato di operazioni da compiere (esattamente due), mentre qui il numero delle operazioni varia col variare del numero, o meglio aumenta con l'aumentare del numero: due alla seconda si calcola con 2×2 , due alla quarta con $2 \times 2 \times 2 \times 2$, e così via.

Ovviamente, vorremmo evitare di cavarcela in questo modo:

```
PER ALLASECONDA :X  
RIPORTA :X * :X  
FINE
```

```
PER ALLATERZA :X  
RIPORTA :X * :X * :X  
FINE
```

```
PER ALLAQUARTA :X ...
```

La soluzione al problema sta nella definizione di potenza, così come viene insegnata a scuola:

- qualsiasi numero elevato allo zero dà uno.

Esercizi - Capitolo tredicesimo

1) Scrivete una procedura Ennesimo :N :Lista che estragga dalla lista data l'elemento desiderato, cioè il terzo, il quarto, eccetera, indicato da :N. Si può ragionare come segue:

- a) il primo elemento di una lista è il Pri;
- b) per numeri più grandi di uno, consideriamo il Mp e sottraiamo uno al numero: per esempio per trovare il quarto elemento di [A B C D E] dobbiamo cercare il terzo di [B C D E]. Anche stavolta esiste già la primitiva corrispondente, che si chiama Elemento.

2) Scrivete una procedura:

```
PER VOCALE? :CARATTERE
```

che riporti Vero se il carattere è una vocale, e Falso altrimenti. Da questa, potrete poi scrivere una procedura Cconsonante? semplicemente così:

```
PER CONSONANTE :CAR  
RI NON VOCALE? :CAR  
FINE
```




• X alla N è uguale a X moltiplicato per X alla N-1 (Per esempio, quattro alla terza è uguale a quattro per quattro alla seconda.

Queste due righe di definizione ci permettono di calcolare ricorsivamente potenze grandi a piacere: la seconda riga riduce l'esponente ad un altro che è più semplice da calcolare, perché è più vicino allo zero, mentre la prima ci permette di fermarci dandoci, appunto, il risultato della funzione quando l'esponente è uguale a zero.

La stessa cosa si può scrivere in Logo, e ha anche un aspetto meno complicato, traducendo (per esempio) quattro alla terza con:

```
POTENZA 4 3:PER POTENZA
:BASE :ESP
SE :ESP = 0 RIPORTA 1
RIPORTA :BASE * POTENZA
:BASE ( :ESP - 1 )
FINE
```

Quando Esp equivale a zero viene eseguita la prima linea; negli altri casi viene eseguita la seconda, che contiene la chiamata ricorsiva. E ogni impresario si ricorda che dovrà moltiplicare per la Base il numero che gli verrà riportato dalla Potenza successiva.

Anche qui il problema della ricorsione si riduce sostanzialmente a due punti: trovare una base (o anche più d'una), che permetta di fermarsi, e trovare un passo che permetta di avvicinarsi alla base. E alla base dobbiamo essere sicuri di arrivarci, prima o poi, altrimenti il programma, come si dice, entra in un ciclo infinito. Questo succede, per esempio, se lanciate la Potenza con un argomento negativo o decimale, perché applicando successivamente il passo, non è possibile raggiungere lo zero.

Comunque, quando abbiamo a che fare con numeri interi, lo ze-

ro è solitamente la base ideale della ricorsione: possiamo infatti considerare tutti i numeri come ottenuti dallo zero, sommando successivamente uno. Tra l'altro è proprio questa la definizione formale di numero data solitamente dai matematici.

Vero o falso?

Esiste in Logo, come del resto in tutti i linguaggi di programmazione, una famiglia di primitive, di cui non abbiamo ancora parlato esplicitamente. Si tratta di tutte quelle che usiamo normalmente nelle istruzioni condizionali, cioè dopo la parola Se; le più comuni riguardo ai numeri sono rappresentate dal simbolo di uguale (=), dal simbolo di maggiore (>) e dal simbolo di minore (<). Se proviamo a digitare semplicemente:

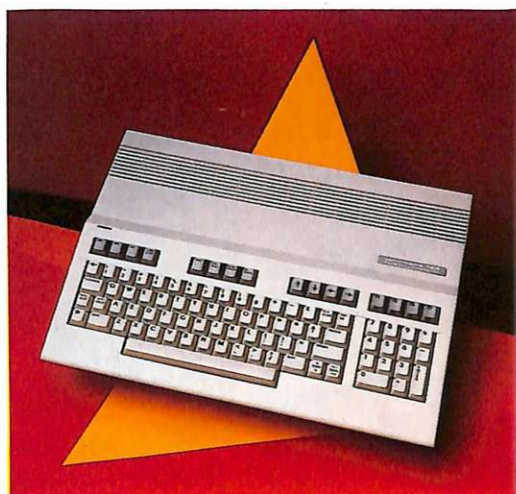
```
5 < 7 e 7 < 5
```

ci accorgiamo che viene riportato un risultato che può essere soltanto Vero o Falso. Le procedure di questo tipo vengono chiamate di solito Predicati, e servono appunto a verificare se una certa condizione è o non è vera.

Detto questo, si comprende meglio il funzionamento di Se, che per verificare la condizione che gli compete, non fa altro che calcolare il suo primo argomento (che deve dipendere da un predicato) e ricevere Vero o Falso in risposta. In altre parole:

```
SE :X = 12 ALLORA ....
```

passa per prima cosa il controllo alla primitiva =, che riporterà Vero nel caso che :X sia effettivamente uguale a 12 e Falso altrimenti. A seconda di questo risul-



tato, quello che segue sulla stessa linea verrà o meno eseguito.

Programmare una di queste procedure-arbitro non è complicato: eccone una che verifica se un numero è positivo:

```
PER POSITIVO? :N
SE :N > 0 RI "VERO (RI è l'abbreviazione
RI "FALSO di RIPORTA)
FINE
```

ed una che controlla se un numero è pari:

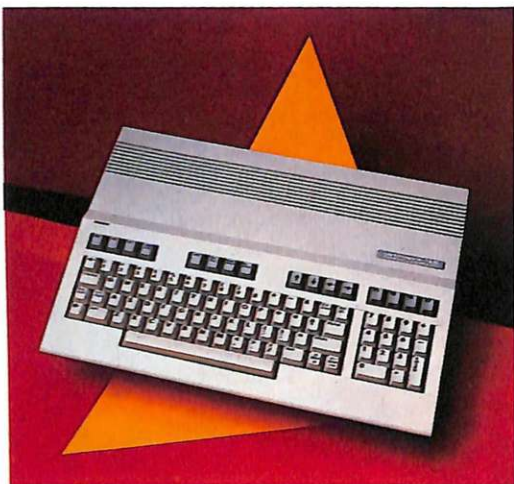
```
PER PARI? :N
SE :N / 2 = INTERO :N / 2 RI
"VERO
RI FALSO
FINE
```

Ovvero, un numero è pari se la sua metà è un numero intero. Queste due ultime procedure si possono anche scrivere, più elegantemente, così:

```
PER POSITIVO? :N
RI :N > 0
FINE
```

```
PER PARI? :N
RI :N / 2 = INTERO :N / 2
FINE
```

Fernando Zanini
(continua)



Capitolo Tredicesimo

Parole e liste

In questo paragrafo esamineremo finalmente con un po' d'ordine tutte le questioni riguardanti virgolette, due punti e parentesi quadre. La struttura delle frasi ammesse in Logo è simile a quella delle frasi italiane all'imperativo le procedure corrispondono al verbo (per esempio, vai avanti!), e i loro argomenti corrispondono al gruppo nominale (per esempio, di cinquanta passi!). L'analogia in effetti regge abbastanza, le procedure indicano azioni da compiere, e sono quindi un po' come i verbi; gli argomenti (che possono anche essere funzioni, cioè procedure che indicano un oggetto) sono gli oggetti che hanno a che vedere con queste azioni; per dare loro un nome in generale li chiameremo appunto oggetti, mentre in italiano possiamo parlare di ogni tipo di oggetti (cose materiali, idee, eccetera), gli oggetti del Logo possono essere di due soli tipi: parole o liste. Vediamoli da vicino.

Parole

Chiamiamo parola una qualunque sequenza di caratteri non separati da spazi. È anche possi-

bile, ma non è molto usato, avere delle parole con dentro spazi o parentesi o altro, basta racchiuderle tra due apici (virgolette singole), come qui sotto:

QUESTA È UNA PAROLA SOLA !!!?(!)(‘

Ad ogni modo, noi useremo sempre parole scritte in modo usuale, cioè senza apici.

La gente, sentendo parlare in italiano, è in grado di comprendere

al volo se una certa parola è un verbo o un nome. Il Logo invece pretende che i vari tipi grammaticali siano scritti in modi diversi. Abbiamo quindi esattamente tre modi di scrivere una parola:

- **Non preceduti da alcun simbolo:** è sicuramente il modo che usiamo di più. Quando il Logo vede una parola che non è preceduta da un simbolo (due punti o virgolette) pensa che si tratti di una procedura, e cerca di eseguirla come tale.

- **Con le virgolette:** le virgolette vanno scritte immediatamente prima della parola (senza spazio), e non occorre scriverle dopo (per esempio "Lato), perché la parola finisce al primo spazio. Come abbiamo visto altrove, scrivere una parola con le virgolette significa riferirsi alla parola stessa come sequenza di caratteri, e non al suo valore o al suo significato.

- **Con i due punti:** essi vanno scritti subito prima della parola (per esempio :Lato). In questo caso la parola deve essere un Nome, cioè deve già avere un qualche valore. Quello a cui ci si riferisce usando i due punti è appunto il Valore del nome, cioè quello che c'è dentro al cassetto, chiamato così. E lì dentro non potrà

esserci altro che una parola o una lista. Ci sono esattamente due modi di conferire un valore ad una parola, o se preferite di mettere un oggetto dentro a un cassetto. Il primo modo è usare il nome del cassetto come parametro di una procedura, dichiarandolo (con i due punti) di seguito, al titolo della procedura stessa.

PER TRIANGOLO :LATO

eccetera. A questo punto ogni volta che chiamerete la procedura Triangolo (per esempio con Triangolo 50) il nome Lato assumerà direttamente il valore dell'argomento (nell'esempio, 50). Questo nome, che è appunto il parametro, sarà una variabile locale, e come tale verrà buttato via alla fine della procedura.

Il secondo modo di assegnare un valore a una parola è usare la primitiva Assegna. Vediamo alcuni esempi:

ASSEGNA "NUMERO 3

assegna al nome Numero il valore 3. D'ora in avanti quando digiteremo :Numero intenderemo 3, e quando digiteremo "Numero intenderemo sempre Numero.

ASSEGNA "RISPOSTA "SI

assegna al nome Risposta il valore Si. Di qui in avanti :Risposta starà per Si.

ASSEGNA "LATO :NUMERO

Ora abbiamo i due punti davanti a Numero, e quindi stiamo parlando del suo valore, che è tre. Così, d'ora in poi Lato varrà tre (come del resto Numero).

ASSEGNA "LATO :LATO + 1

Anche questo è legittimo, e in fondo perfettamente logico: asse-



gnamo al nome Lato il suo precedente valore aumentato di uno. Adesso Lato varrà dunque quattro.

ASSEGNA :RISPOSTA :NUMERO

Che cosa succede qui? Il valore di Numero, che è ancora 3, viene assegnato al valore di Risposta, che è Si. Perciò d'ora in poi premendo :Si intenderemo 3; il valore di Risposta non verrà per niente modificato.

ASSEGNA "LYSTA [A B C D]

Infine, dopo questo comando il nome Lysta avrà come valore [A B C D] che, come vedremo tra poco, è una lista.

• Numeri

I numeri meritano un breve discorso a parte. Essi sono conside-

rati dal Logo parole a tutti gli effetti; l'unica differenza sostanziale rispetto alle altre parole è che vi si possono anche fare operazioni aritmetiche.

Tuttavia, sebbene siano oggetti, non li scriviamo con le virgolette né con i due punti (anche se con le virgolette potremmo farlo). Il Logo funziona così soltanto per comodità, dal momento che i numeri vengono usati molto spesso. In effetti, è come se essi fossero procedure definite pressappoco così:

```
PER 1
RI "1
FINE
PER 2
RI "2
FINE
PER 3 .....
```

Questo non è comunque quello che succede in realtà; il linguag-

gio Logo non vi permetterà mai di usare un numero come nome di una procedura.

• Liste

Una lista è una sequenza di oggetti (parole o altre liste) racchiusa all'interno delle parentesi quadre. Per esempio:

[QUESTA È UNA LISTA]

è una lista che contiene quattro elementi, ognuno dei quali è una parola. Ma anche:

ANCHE [QUESTA [È UNA]] LISTA]

è una lista, che però contiene tre elementi: il primo è una parola, Anche; il secondo è una lista, [Questa [È Una]]; e il terzo è di nuovo una parola, Lista. La lista di mezzo, [Questa [È Una]], è poi composta da due elementi, e così via. Tutto ciò è un po' noioso; la sua utilità rivela non appena abbiamo bisogno di tenere assieme dei dati in un qualche ordine.

Un esempio dei più classici è il problema di archiviare i nomi dei nostri amici. La prima pensata è certo quella di mettere i nomi in una lista, così:

[MARCO GIOVANNI ANTONIO]

Se vogliamo inserire i cognomi, bisogna che arrangiamo la cosa per far sì che ogni cognome sia unito al nome corrispondente, e possiamo usare una lista di liste:

[MARCO GRASSI] [GIOVANNI ROSSI] [ANTONIO SALTAFOSSI]

In questo modo ogni elemento della lista consiste in un'altra lista con nome e cognome della stessa persona. A questo punto, potremmo sostituire a ciascuna delle sottoliste come:

Tavola 1.

```
PER AGGIUNGI :NOMELISTA
MOSTRA COSA :NOMELISTA
ST [SCRIVI LA PAROLA DA AGGIUNGERE]
ST [(OPPURE "STOP" PER USCIRE)]
ST [(OPPURE "T" PER TOGLIERE)]
LOCALE "NUOVAPAROLA
AS "NUOVAPAROLA PRI LR
SE :NUOVAPAROLA = "STOP STOP
SE :NUOVAPAROLA = "T TOGLI :NOMELISTA STOP
AS :NOMELISTA INPRI :NUOVAPAROLA COSA :NOMELISTA
AGGIUNGI :NOMELISTA
FINE
```

```
PER TOGLI :NOMELISTA
MOSTRA COSA :NOMELISTA
ST [SCRIVI LA PAROLA DA TOGLIERE]
ST [(OPPURE "STOP" PER USCIRE)]
ST [(OPPURE "A" PER AGGIUNGERE)]
LOCALE "PAROLA
AS "PAROLA PRI LR
SE :PAROLA = "STOP STOP
SE :PAROLA = "A AGGIUNGI :NOMELISTA STOP
AS :NOMELISTA SENZA :PAROLA COSA :NOMELISTA
TOGLI :NOMELISTA
FINE
```




[GIOVANNI ROSSI]

una lista che contenga più informazioni, per esempio:

[GIOVANNI ROSSI] [VIA GARI-
BALDI 99] [AOSTA]]

e ASSEGNARE tutto quanto a un nome, diciamo Amici, che conterrà tutti i dati che ci servono in perfetto ordine.

Resta però un problema, e non piccolo, che costituisce in parte l'essenza stessa della programmazione: non basta inserire i dati, bisogna anche saperli estrarre. La lista che avremmo ottenuto, così com'è scritta, non è infatti molto più utile di un'agenda, anzi è molto meno leggibile.

Lo scopo principale di questo capitolo e dei prossimi, è appunto quello di imparare a maneggiare in questo senso le liste, scrivendo procedure che, su semplice richiesta, vadano a estrarre proprio i dati che ci interessano, senza confusione.

Queste possibilità di manipolazione ci permetteranno tra le altre cose di scrivere programmi amichevoli che spieghino il loro stesso funzionamento, e di fare qualche piccolo esperimento sulla lingua italiana.

Un po' di primitive

Il nostro problema ora è quello di andare dentro alle liste e di poterle leggere o modificare a nostro piacimento. Ripassiamo brevemente le primitive che ci serviranno allo scopo.

L'operazione principale che ci permette di scomporre una lista consiste nell'estrarne il primo elemento. Per farlo usiamo la primitiva Primo (abbreviata Pri):

PRIMO [CARLO MARIA GIUSEPPE ANTONIO]

riporta il risultato Carlo. Il resto

della lista (che è ancora una lista, ma senza il primo elemento), è chiamato in Logo Menpri (abbreviato Mp):

MENPRI [CARLO MARIA GIUSEPPE ANTONIO]

riporterà [Maria Giuseppe Antonio].

Analogamente funzionano le primitive Ultimo e Menult (abbreviate rispettivamente Ult e Mu). Potete provarle su qualunque lista. Le primitive inverse a queste, quelle in grado di mettere un oggetto rispettivamente al primo e all'ultimo posto di una lista data, sono InPri e InUlt. Entrambe vogliono due argomenti che saranno l'elemento da inserire e la lista, come in:

INPRI "AUGUSTO [NERONE
COSTANTINO]
INULT "LINCOLN [WASHINGTON
EISENHOWER]

Se usate una lista come primo argomento di queste due primitive, come in:

INPRI [SPAGHETTI MAC-
CHERONI] [TAGLIATELLE
POLENTA]

noterete che essa viene presa tale e quale, e infilata dentro l'altra con tanto di parentesi.

La primitiva Frase (abbreviata Fr), serve a unire tra loro due liste, due parole o una parola e una lista, e non lascia parentesi intermedie.

FRASE [PIERINO MANGIA]
[LA MELA]
FRASE "ROSSO "VERDE
FRASE "AZZURRO [È IL CO-
LORE DEI TUOI OCCHI]
(FRASE [QUESTA] [HA TRE]
"ARGOMENTI)

Nell'ultimo esempio, le parentesi tonde servono a far accetta-

re a Frase un numero di argomenti diverso da due.

Lo spazio che c'è prima della parentesi chiusa è necessario per evitare che il Logo consideri la parentesi come facente parte dell'ultima parola, che diventerebbe "Argomenti).

Ci sono altre primitive con cui potete usare le parentesi tonde allo stesso modo, tra cui Stampa, Parola e Lista.

Parola appunto prende i suoi argomenti (normalmente due, che devono essere parole) e li unisce in una sola parola:

PAROLA "QUARANTA "QUAT-
TRO
(PAROLA "NON "TI "SCORDAR
"DI "ME)

e così via. La primitiva Lista prende i suoi argomenti e li mette dentro a una lista che costruisce appositamente, come potete verificare con:

LISTA ANDREA PAOLO
LISTA [GIORGIO] [AMEDEO]
(LISTA 1 2 [3 3.5 3.8] 4)

Un'ultima cosa: Primo, MenPri, Ultimo e MenUlt funzionano anche con le parole; cercate di scoprire da soli come, se già non ve lo immaginate.

Questo mi ricorda qualcosa

Usando ripetutamente le primitive Pri e Mp (abbreviazioni di Primo e MenPri) possiamo passare un'intera lista: basta che, dopo aver esaminato il primo, lo tagliamo via, ovvero consideriamo il Mp e ricominciamo da capo. Il Pri della nuova lista sarà il secondo della lista originale, e così via.

Che cosa succede quando nella lista non rimane più niente? Potete scoprirlo chiedendo il Mp di una lista di un solo elemento, per esempio:

MP [MARIA]

Il risultato è [], una lista vuota. Ora, possiamo usare la lista vuota proprio come usavamo lo zero per i numeri naturali: come sottraendo continuamente uno sapevamo di raggiungere prima o poi lo zero, così qui siamo sicuri che continuando a togliere il Pri a una lista, riusciremo a raggiungere la lista vuota.

Proviamo a scrivere una procedura ricorsiva, che calcoli la lunghezza di una lista, cioè di quanti elementi la lista è composta. Per cominciare, sappiamo che la lista vuota ha lunghezza zero:

```
PER LUNGHEZZA :LISTA
SE :LISTA = [] RIPORTA 0
```

o anche:

```
PER LUNGHEZZA :LISTA
SE VUOTO? :LISTA RI 0
```

Vuoto? è un predicato che useremo spesso, e che ci dice se una lista (o una parola) è o no uguale alla lista vuota (o alla parola vuota, che vedremo presto).

Una volta che si è stabilita la condizione di base, dobbiamo solo spiegare al Logo come calcolare la lunghezza di una lista conoscendo quella del suo MenPri, che è più corta. Basterà prendere la lunghezza del MenPri e aggiungere uno.

```
PER LUNGHEZZA :LISTA
SE VUOTO? :LISTA RI 0
RI 1 + LUNGHEZZA MP :LISTA
FINE
```

Lunghezza [A B C D E F] dovrebbe ora riportare la lunghezza della lista, che è sei. Se il suo funzionamento non vi è del tutto chiaro, potete seguirlo attentamente con la Traccia attiva. Tracciamo comunque insieme una chiamata abbastanza semplice come:

LUNGHEZZA [A B]

Gli impresari chiamati in causa sono tre:

1) Il primo deve riportare (a voi) la lunghezza di [A B]. Dal momento che [A B] non è vuota (prima riga), egli si limita (seconda riga) ad aggiungere uno al risultato che gli verrà riportato dall'impresario Lunghezza [B], essendo [B] il MenPri della lista originale.

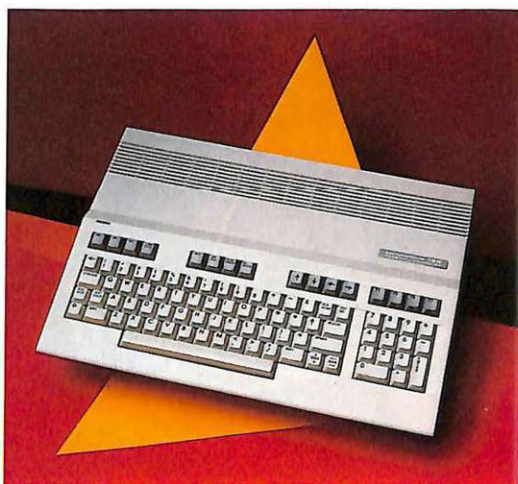
2) Il secondo impresario deve quindi calcolare la lunghezza di [B] e comunicarla (riportarla) al primo affinché questi possa finire. La lista non è vuota (prima riga, di nuovo), e il nostro omino sa solo che dovrà aggiungere uno a Lunghezza [], dato che la lista vuota [] è il Mp della lista [B].

3) Il terzo impresario non è quindi altri che Lunghezza [], ed ha decisamente vita facile: il suo argomento è la lista vuota, e la prima riga della definizione gli dice in questo caso di riportare zero. Egli telefona quindi, al secondo impresario, che lo aveva chiamato, e gli dice: Zero!

2) Il secondo impresario, che stava aspettando una risposta da prima, appena riceve il numero aggiunge uno e dice al primo impresario: Uno!

1) Ormai sappiamo come va a finire, il primo impresario riceve finalmente il messaggio, aggiunge uno e risponde a chi lo aveva chiamato... Un momento, chi lo aveva chiamato?... Ah, eravamo stati noi... Ci risponde "Due"!, che è per l'appunto la lunghezza della lista [A B].

Tutto questo lavoro è stato decisamente inutile a fini pratici: si dà il caso infatti che esista una primitiva Logo, chiamata Conta, che funziona esattamente come



la nostra Lunghezza. Una cosa forse vi meravighierà; provate a digitare:

LUNGHEZZA "ARCIVESCO-
VODICOSTANTINOPOLI

Conta produce lo stesso effetto) e scoprirete che la vostra procedura è anche in grado di contare le lettere di una parola.

Vediamo come ciò possa succedere: abbiamo detto che Pri e Mp funzionano anche su parole, riportando rispettivamente la prima lettera e tutto il resto. Ad un certo punto, continuando a "tagliare la testa" alla parola, una qualche procedura si troverà come argomento una parola vuota, che viene riconosciuta dal predicato Vuoto?.

Tutto qui: il resto funziona come con le liste. Se volete scrivere esplicitamente la parola vuota, potete farlo così:

E potete anche ASSEGNARle un valore!

Predicato su parole e liste

Per quanto riguarda i predicati, non c'è niente di particolarmente nuovo da dire.

Abbiamo già visto Vuoto?, e certamente sarà utile sapere che



esistono predicati che sanno riconoscere una lista, una parola oppure un numero.

Essi si chiamano rispettivamente Lista?, Parola? e Numero? (con il punto interrogativo).

L'abitudine di mettere un punto interrogativo in fondo al nome dei predicati (anche a quelli definiti da voi) è particolarmente lodevole, dal momento che permette di riconoscerli immediatamente; dopo tutto lanciare un predicato è come fare una domanda la cui risposta sarà sì o no.

Un altro predicato particolarmente utile è quello che ci dice se un certo oggetto appartiene o meno a una lista.

APPARTIENE? "MILANO [ROMA MILANO FIRENZE]
APPARTIENE? "MILANO [LOMBARDIA VENETO PIEMONTE]
AP? "MILANO [[MILANO TORINO] [ROMA FIRENZE]]

(Ap? è l'abbreviazione di Appartiene?). La prima affermazione è vera, le altre due sono false. Nella terza linea, la lista ha solo due elementi, [Milano Torino] e [Roma Firenze], e nessuno dei due è uguale a Milano. Appartiene? si comporta esattamente come se fosse stato definito così:

```
PER AP?:ELEM
:LISTA
SE VUOTO? :LISTA
RIPORTA "FALSO
SE PRI:LISTA =
:ELEM RIPORTA
"VERO
RIPORTA AP?:ELEM
MENPRI :LISTA
FINE
```

cioè, continuando a tagliare la testa alla lista, prima o poi si verificherà uno di questi due casi: o l'elemento cercato è al primo posto nella lista (e allora sarà

Vero che c'è), oppure la lista resta vuota (nel qual caso evidentemente l'elemento non c'era).

Capitolo Quattordicesimo

Un piccolo generatore di frasi

In questo capitolo cercheremo di abbozzare un programma che generi delle frasi italiane molto semplici pescando parole a caso (anzi, ACaso) da liste preparate appositamente.

Avremo bisogno di una procedura che estragga un elemento a caso da una lista data, e la chiameremo Estrai.

Poi ci serviranno delle liste che contengano parole raggruppate secondo categorie grammaticali, cioè nomi, verbi, eccetera. Cominciamo a preparare queste ultime; metteteci pure i nomi dei vostri amici.

```
AS "NOMIPROPRI [MARCO
MARIA GIOVANNI SARA]
AS "VERBI [AMA MANGIA
ODIA CORTEGGIA AGGREDISCE]
```

D'ora in poi, il nostro programma per generare delle frasi potrebbe funzionare in questo modo: prendere un nome pro-

prio, un verbo e un altro nome proprio e riunirli in una Frase:

```
PER GENERAFRASE
RIPORTA (FRASE NOME VERBO NOME)
FINEPER NOME
RI ESTRAI :NOMIPROPRI
FINEPER VERBO
RI ESTRAI :VERBI
FINE
```

Ora, tutto quello che ci manca è definire la procedura Estrai. Possiamo utilizzare in essa la primitiva Elemento, che permette di estrarre un elemento prescelto:

ELEMENTO 3 :NOMIPROPRI

Essa riporterà Giovanni, che è il terzo elemento della lista dei nomi propri. Dobbiamo però sostituire a tre un numero scelto a caso ogni volta. Ora, che argomento dare alla funzione ACaso? Il numero estratto non potrà essere più grande della lunghezza della lista (non possiamo chiedere il quinto elemento di una lista lunga quattro); d'altra parte, se stabilissimo un numero fisso, dovremmo usare tutte liste di uguale lunghezza. C'è però un modo per cavarsela: sappiamo che:

CONTA :NOMIPROPRI

dà la lunghezza della lista; quindi:

ACASO (CONTA :NOMIPROPRI)

darà un numero compreso tra zero e Conta :Nomipropri meno uno (per esempio, se la lista è lunga sette, darà un numero tra zero e sei). Aggiungiamo uno:

1 + ACASO (CONTA :NOMIPROPRI)

e otteniamo un numero casuale compreso tra zero e la lunghezza



della lista, che è esattamente quello che ci serviva.

A questo punto possiamo estrarre l'elemento corrispondente a quel numero:

ELEMENTO 1 + ACASO (CONTA :NOMIPROPRI) :NOMIPROPRI

e costruire così la procedura Estrai:

PER ESTRAI :LISTA
RI ELEMENTO 1 + ACASO
(CONTA :LISTA) :LISTA
FINE

Questo dovrebbe essere sufficiente perché:

STAMPA GENERAFRASE

faccia il suo dovere.

Un programma di inserimento

Se ora volete aggiungere delle parole alle vostre liste, sarete costretti a usare di nuovo Assegna (potreste anche usare l'editore; se non sapete come, consultate l'appendice B). Non sarebbe male invece scrivere una procedura che aggiunga a una certa lista delle parole ricevute direttamente da tastiera. Ovviamente, questa procedura dovrà avere una ASSEGNAzione al suo interno, altrimenti non potrebbe cambia-

re :NomiPropri o :Verbi che sono variabili globali. Per prima cosa la procedura dovrà chiederci a quale lista aggiungere i dati:

PER INSERIMENTO
ST [QUALE LISTA VUOI MODIFICARE?]

Per leggere qualcosa dalla tastiera usiamo LeggiRig (abbreviato Lr), che però riporta una intera linea (finché non premete Return) e la racchiude all'interno di una lista. Potete verificarlo premendo:

LR

e poi qualsiasi serie di sciocchezze. Dal momento che a noi interessa una sola parola, prendiamo Primo LeggiRig, lo mettiamo in una variabile locale e chiamiamo una procedura Aggiungi, che programmeremo per fare tutto il resto.

PER INSERIMENTO
ST [QUALE LISTA VUOI MODIFICARE?]
LOCALE "NOMELISTA
AS "NOMELISTA PRI LR
AGGIUNGI :NOMELISTA
FINE

Questa procedura si può anche scrivere così:

PER INSERIMENTO
ST [QUALE LISTA VUOI MODIFICARE?]
AGGIUNGI PRI LR
FINE

Il primo compito di Aggiungi sarà sicuramente quello di chiederci una parola e metterla in una variabile locale.

PER AGGIUNGI :NOMELISTA
ST [SCRIVI LA PAROLA DA AGGIUNGERE]
LOCALE "NUOVAPAROLA
AS "NUOVAPAROLA PRI LR

Esercizi - Capitolo quattordicesimo

1) *Esandete la procedura Nome in modo che essa scelga a caso se estrarre un nome proprio o qualcos'altro, per esempio un nome comune da una lista che preparerete. Per farlo potete usare una condizione come:*

SE ACASO 3 = 2 fai qualchecosa

così il seguito della linea sarà eseguito circa una volta su tre.

2) *La procedura Senza è in grado di togliere soltanto una occorrenza di un elemento da una lista. Per esempio:*

SENZA 6 [3 7 6 5 6 4 6 6]

eliminerebbe soltanto il primo 6, senza raggiungere gli altri. Scrivete una procedura che invece riporti la lista privata di Tutte le occorrenze dell'oggetto.

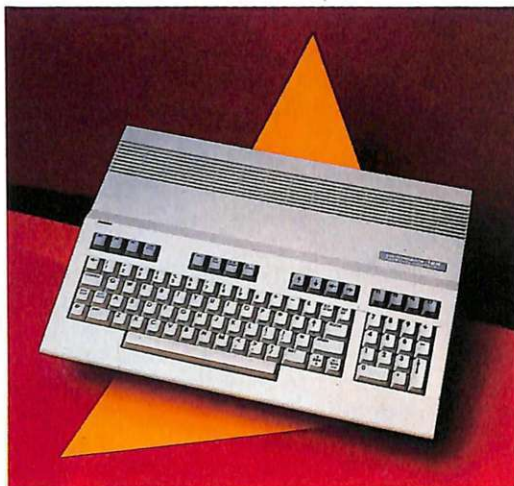
Scrivete una procedura:

SOSTI :NUOVO :VECCHIO :LISTA

che riporti il risultato della SOSTituzione di :Nuovo a :Vecchio nella lista. Per esempio:

SOSTI "PEDALATO "CAMMINATO [ANTONIO HA CAMMINATO TUTTO IL GIORNO]

dovrebbe riportare : [ANTONIO HA PEDALATO TUTTO IL GIORNO]



Ora facciamo l'assegnazione: dobbiamo assegnare al nome contenuto nella variabile NomeLista (che indicheremo quindi con i due punti) il valore precedente di quel nome (la vecchia lista) con in più la parola contenuta in NuovaParola (che indichiamo ugualmente con i due punti).

La faccenda è più intricata di quanto sembri, e richiede l'impiego di una nuova primitiva, Cosa. Vediamo perché.

Per prima cosa supponiamo di chiamare Inserimento e di comunicargli che vogliamo modificare la lista Verbi, che consiste in: [Ama Mangia Odia Corteggia Aggradisce].

A questo punto entra in azione Aggiungi e noi decidiamo di aggiungere il verbo Ripudia.

È chiaro che l'operazione che va fatta non è altro che:

```
AS "VERBI INPRI "RIPUDIA
:VERBI
```

ossia prendere la lista dei verbi, inserire Ripudia e rimettere tutto nel cassetto chiamato Verbi.

Ora, Ripudia è il valore di NuovaParola, per cui di seguito possiamo scrivere:

```
AS "VERBI INPRI :NUOVAPAROLA
:VERBI
```

Il problema rimasto è sostituire Verbi, che è il valore di NomeLista; "Verbi (con le virgolette) si tradurrà con :NomeLista (essendo appunto il suo valore) mentre il valore di Verbi, e qui sta la novità, si può scrivere Cosa :NomeLista. Che è come dire il valore del valore di NomeLista, o la Cosa che sta nella Cosa che è dentro NomeLista.

Di fatto, la primitiva Cosa equivale esattamente ai due punti, solo che il Logo non ci permette di scrivere per due volte consecutive questi ultimi. Riassumendo, quindi, la nostra procedura diventa:

```
PER AGGIUNGI :NOMELISTA
ST [SCRIVI LA PAROLA DA
AGGIUNGERE]
LOCALE "NUOVAPAROLA
AS "NUOVAPAROLA PRI LR
AS :NOMELISTA INPRI :NUOVAPAROLA
COSA :NOMELISTA
AGGIUNGI :NOMELISTA
FINE
```

La chiamata ricorsiva permette di aggiungere più nomi alla volta all'interno della stessa lista. Ultima pignoleria: per fermare questo programma siamo costretti a premere Ctrl-G.

Che cosa possiamo escogitare per farlo invece terminare "naturalmente"?

L'unico modo possibile è agire sulla parola che inseriamo: possiamo inoltre chiedere alla procedura di fermarsi quando legge, per esempio, Stop.

Per rendere tutto più chiaro, specifichiamo la faccenda stampando un messaggio promemoria. Inoltre, chiediamo di mostrarci ogni volta il contenuto della lista, per averlo sempre

sott'occhio. Mostra è come Stampa, solo che fa vedere anche le parentesi esterne delle liste.

```
PER AGGIUNGI :NOMELISTA
MOSTRA COSA :NOMELISTA
ST [SCRIVI LA PAROLA DA AGGIUNGERE]
ST [(OPPURE "STOP" PER USCIRE)]
LOCALE "NUOVAPAROLA
AS "NUOVAPAROLA PRI LR
SE :NUOVAPAROLA = "STOP
STOP
AS :NOMELISTA INPRI :NUOVAPAROLA
COSA :NOMELISTA
AGGIUNGI :NOMELISTA
FINE
```

Questa versione di Aggiungi non si può abbreviare come avevamo fatto poco fa con Inserimento, perché la parola letta da tastiera viene usata più di una volta: se la indicassimo sempre con PRI LR il Logo si aspetterebbe che la rigiditassimo ogni volta.

Un programma di rimozione

Vi è per caso successo di inserire qualcosa per sbaglio in una lista usando Inserimento? Oppure avete cambiato idea riguardo a qualche parola inserita e ora la volete togliere?

Per poterlo fare con comodo scriveremo una nuova procedura che sia in grado di togliere un certo elemento da una lista; possiamo tentare di ricalcarla da Aggiungi.

```
PER TOGLI :NOMELISTA
MOSTRA COSA :NOMELISTA
ST [SCRIVI LA PAROLA DA TOGLIERE]
ST [(OPPURE "STOP" PER USCIRE)]
LOCALE "PAROLA
AS "PAROLA PRI LR
SE :PAROLA = "STOP
STOP
AS :NOMELISTA SENZA :PA-
```




ROLA COSA :NOMELISTA
TOGLI :NOMELISTA
FINE

L'unica differenza rilevante rispetto ad Aggiungi è nel punto in cui effettivamente la parola viene tolta: abbiamo usato una procedura Senza che è ancora da definire.

PER SENZA :ELEMENTO
:LISTA

Essa prenderà due argomenti, un oggetto e una lista, e dovrà riportare una lista uguale all'originale ma Senza l'oggetto. Purtroppo non è assolutamente detto che l'oggetto sia al primo posto (altrimenti potremmo usare Mp) e al limite non è nemmeno detto che l'oggetto ci sia davvero (nel qual caso il risultato dovrà essere la lista intatta).

Proviamo a pensarla in maniera ricorsiva, considerando ogni volta soltanto il Pri della lista. Si possono verificare solamente tre casi.

1) La lista è vuota. Allora sicuramente non contiene l'oggetto, quindi dobbiamo riportare la lista stessa.

SE VUOTO? :LISTA RI :LISTA

2) Il PRI della lista è uguale all'oggetto da eliminare. Dal momento che l'oggetto è al primo posto, la lista Senza l'oggetto sarà uguale al MP della lista:

SE PRI :LISTA = :ELEMENTO
RI MP :LISTA

3) Tutti gli altri casi, cioè la lista non è vuota e al primo posto c'è un oggetto che non dobbiamo togliere. Il trucco dovrebbe essere ormai familiare: chiediamo a qualcun altro di riportarci il Menpri della lista Senza l'elemento, e tutto quello che dobbia-

mo fare noi sarà rimettere il Pri davanti al risultato.

RI INPRI PRI :LISTA SENZA
:ELEMENTO MP :LISTA

Riassumendo:

PER SENZA :ELEMENTO
:LISTA
SE VUOTO? :LISTA RI :LISTA
SE PRI :LISTA = :ELEMENTO
RI MP :LISTA
RI INPRI PRI :LISTA SENZA
:ELEMENTO MP :LISTA
FINE

Verificate il funzionamento di Senza digitando cose come:

SENZA 3 [1 2 3 4 5]
SENZA "NON[SICURAMENTE
OGGI NON PIOVERÀ]

Per finire, possiamo tentare di modificare la procedura Aggiungi per permetterle di chiamare Togli direttamente senza dover fermare il programma. Per farlo aggiungiamo una parola chiave (allo stesso modo di Stop) su cui a ogni giro effettuiamo una verifica. E nulla ci impedisce di modificare anche Togli per permetterci di tornare da lì ad Aggiungi (vedi **tavola 1**). Ora la procedura Inserimento permette sia di aggiungere sia di togliere parole dalle varie liste, ed è molto semplice da usare.

Alcuni consigli pratici

Questo breve paragrafo è dedicato in particolare a chi volesse proseguire sul progetto di generazione di frasi casuali, inserendo costruzioni grammaticali più complicate. Qui parleremo comunque soltanto di frasi piuttosto semplici, al singolare, senza proposizioni subordinate.

Il modo di suddividere una frase, che è stato usato poco fa (nome - verbo - nome) non è certo il più

comodo né il più naturale. Solitamente le proposizioni semplici vengono suddivise in gruppo nominale e gruppo verbale; il gruppo nominale si divide a sua volta in articolo, nome ed eventualmente aggettivi; il gruppo verbale in verbo, eventuali avverbi e gruppo nominale (sempre che il verbo sia transitivo).

Il vostro programma potrà quindi rispecchiare questa struttura, per esempio iniziando così:

PER GENERAFRASE
RI FR GRUPPONOMINALE
GRUPPOVERBALE
FINE

Il gruppo nominale costituisce una discreta noia, perché richiede come minimo di concordare gli articoli. Aiutatevi scrivendo predicati che riconoscano le parole che iniziano con vocale, esse impura, "gn" e zeta, e anche predicati Maschile? :Nome e Femminile? :Nome, che riconoscano il genere dei nomi guardando l'ultima lettera. Per il momento, ovviamente, evitate di usare nomi irregolari (maschili che finiscono in a, per esempio); in seguito si potrebbero disporre i nomi irregolari in apposite liste da controllare ogni volta che occorre conoscere il genere di una parola.

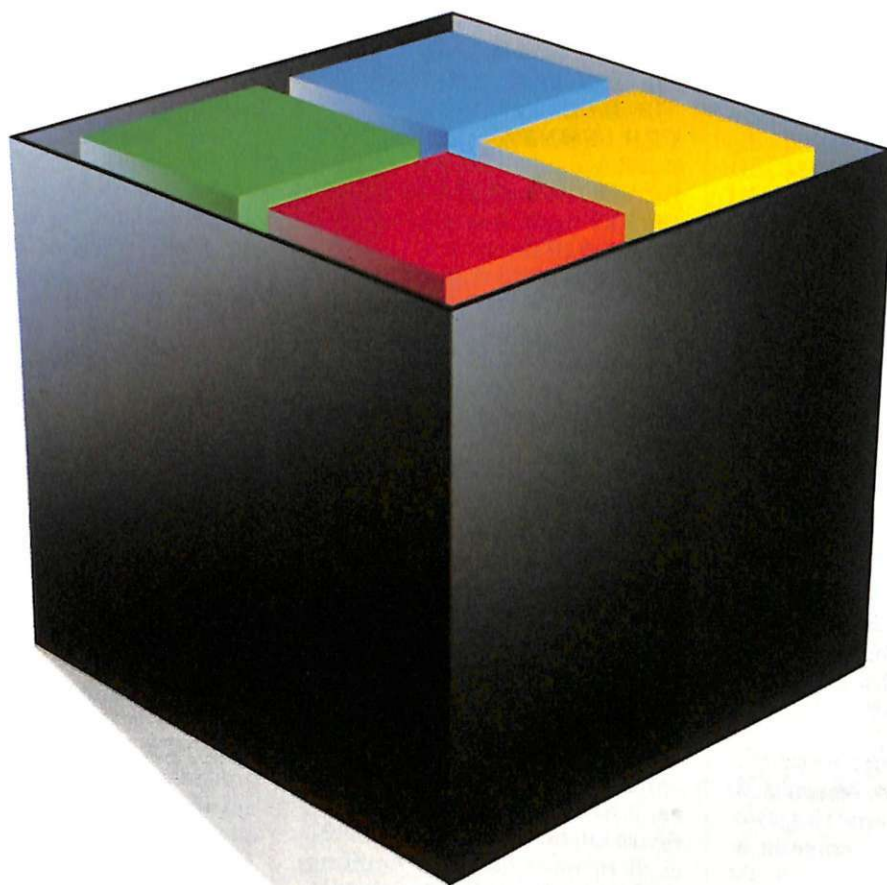
Per quanto riguarda il gruppo verbale, se usate soltanto verbi transitivi potete definirlo così:

PER GRUPPOVERBALE
RI FR VERBO GRUPPONOM
FINE

Altrimenti, bisognerà che il gruppo nominale venga incluso soltanto quando il verbo usato è transitivo. Conviene disporre i verbi transitivi in una lista apposita, così per riconoscerli basterà guardare se sono effettivamente contenute in essa.

Luigi Ambrosi
(continua)

FATE IL VOSTRO GIOCO



Giochi di abilità,
sperimentazione, intuito,
giochi di calcolo,
coordinazione, inventiva.
Partite che si giocano in
ogni ambito professionale,
occasioni da costruire
con regole precise
e con creatività.

Nasce da qui l'impegno
del Gruppo Editoriale JCE
nei settori:

- Informatica,
- Elettronica,
- Comunicazione,
- Economia & Finanza.

Argomenti che richiedono
sempre più competenza,

aggiornamento, e continui
confronti di esperienza.
Ecco la scelta della
massima specializzazione.
Quattro divisioni,
venticinque riviste,
migliaia di notizie
sui giochi del pensiero
innovativo.

L'INFORMAZIONE QUALIFICATA

Sul numero scorso abbiamo visto come mettere su carta la versione definitiva di contratti a uso abitativo con l'aiuto di una qualsiasi stampante. Chiudendo l'argomento, questo mese pubblichiamo un programma anche per contratti a uso non abitativo per coprire tutte le esigenze dei nostri affezionati lettori

Come affittare per non abitare

Questo programma permette di definire un adatto formulario per la creazione, tramite stampante, di contratti di locazione a uso non abitativo, secondo quanto prevede la legge numero 392 del 1978 (unitamente alle modificazioni a questa apportate negli anni seguenti). Una destinazione d'uso immobiliare è di tipo non abitativo, quando il conduttore, cioè la parte che riceve (e paga) l'affitto, usa la casa a fini diversi





da quello dell'abitarvi residenziale: nella pratica gli esempi sono innumerevoli: si va dagli esercizi commerciali agli uffici, dai laboratori agli studi professionali, dalle fabbriche ai bar e ristoranti, dalle sedi di associazioni ai box per auto.

I documenti che si possono ottenere sono direttamente utilizzabili dalle parti che lo devono sottoscrivere (locatore e conduttore) legalmente il supporto cartaceo è una scrittura privata avente validità di accordo formale tra un soggetto che concede un immobile ad uso esclusivamente non abitativo, e un altro soggetto che lo prende in affitto, adeguandosi a una nutrita serie di condizioni esplicitamente indicate.

Equo Canone 3 è adatto a ricevere in input valori e specifiche eventualmente già trattati con software specifico (comparso sul numero di luglio/agosto) per il calcolo dell'ammontare base annuo, in lire, dell'equo canone che il conduttore deve corrispondere al locatore.

Una volta ottenuta la stampa delle copie di contratto necessarie (solitamente ne occorrono tre, in quanto una serve per l'Ufficio del registro) basta che locatore e conduttore, dopo aver preso visione dei contenuti, le sottoscrivano con doppia firma leggibile.

Equo Canone 3 è insomma un utile strumento di lavoro, ideale non solo per geometri, stimatori, ingegneri e agenzie immobiliari,

ma anche per qualsiasi privato, ditta o società che abbiano interesse ad affittare un immobile senza dover interpellare professionisti o comprare contratti-tipo precompilati, spesso troppo generici per i singoli casi. All'utente viene semplicemente chiesto di specificare o indicare, tramite i tasti numerici o di funzione, alcuni dati informativi di obbligatoria citazione (richiesti per legge), tra cui: luogo e data di redazione del contratto, destinazione d'uso esclusivo dell'immobile, generalità del locatore, del conduttore e dell'immobile da locare, durata dell'accordo, canone annuo stabilito, oneri di spesa accessori previsti e modalità di pagamento scadenzato al locatore.

Su questi input il computer lavora velocemente, definendo volta per volta un formulario *ad hoc*, e trasmettendo alla stampante (quella di riferimento è la Commodore MPS 1200) il messaggio software per la produzione del documento cartaceo, che avviene su normali fogli formato A4 (questi diventano sempre tre).

È importante ricordare che il programma ha validità per i soli contratti di locazione a uso non abitativo, e sono quindi esclusi tutti quelli diversi (per esempio le locazioni di case, appartamenti o ville), anche perché è la legge stessa a cui si fa riferimento che prevede le distinzioni delle varie tipologie di destinazione immobiliare.

Tramite un'unica pagina grafica multicolor, si può controllare con facilità lo svolgimento di tutte le routine di elaborazione, con monitoraggio effettuato da un multidisplay che permette una lettura istantanea e soprattutto facilità di interpretazione: il tutto senza possibilità di errori.

La carica sul Commodore 64 si effettua con i consueti comandi di Loading, ovvero Load oppure Load Equo Canone2, o anche con il pratico metodo di pressione contemporanea dei tasti Commodore e Run/Stop, che causa tra l'altro l'Autostart immediato a fine carica, evitando la necessità del Run attivatore.

In memoria RAM il programma occupa uno spazio di quasi 17 Kb, che aumentano di circa 0,7 Kb a lancio avvenuto.

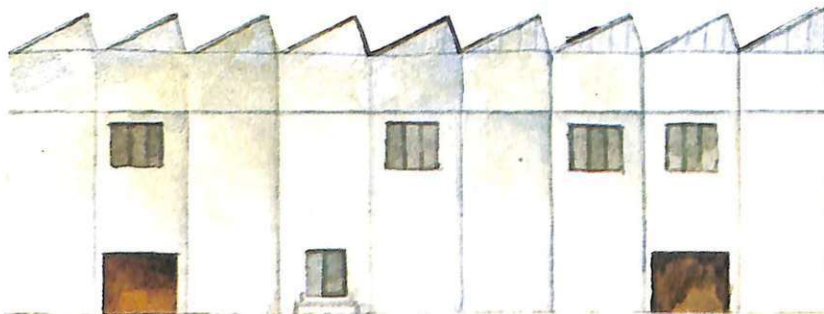
Funzionamento

Terminata l'operazione di caricamento, trascorsi pochi secondi da quando il programma viene fatto partire (tramite un Run di lancio) compare la pagina-monitor a fondo e bordo di colore blu, che rimane invariata nella sua parte grafica e strutturale durante tutte le fasi di esecuzione delle routine.

Nella zona superiore dello schermo sono presenti la grande scritta Equo Canone 3 (nei colori giallo, nero, bianco e verde) e il data-set di copyright (in colore azzurro Reverse).

Un sottotitolo in grigio Reverse evidenzia la funzione del software, che è quella di permettere un'immediata definizione dei formulari adatti a qualsiasi contratto di locazione a uso non abitativo, con stampa dei relativi documenti.

Nella parte inferiore dello schermo, evidenziato da una cornice bianca, trova sistemazione il multidisplay per la scrittura dei messaggi-guida, per gli input e



per la proiezione della videata finale di stampa (in Standby).

Contenuti e riferimenti legislativi del programma, nonché procedure di input ed elaborazione dei dati, fanno riferimento alle disposizioni del Codice civile (in particolare agli articoli: 1456, 1584, 1609 e 1756), alla legge detta "dell'equo canone" numero 392 del 27 luglio 1978 (articoli: 10 e 32 in particolare) e successive modificazioni, nonché naturalmente alla vastissima legislazione vigente in materia di diritto privato, e agli usi e consuetudini locali sulle locazioni.

Anche se non esplicitamente precisato sul video, si intende che l'uso del programma Equo Canone 3 non comporta, comunque e da chiunque venga usato, alcuna responsabilità per chi lo ha realizzato, prodotto e pubblicato; occorrerà pertanto controllare sempre attentamente ogni operazione che porta poi alla stampa del contratto.

Una scritta iniziale lampeggiante in vari colori avverte che la routine esecutiva degli Input può essere avviata solo premendo il tasto Return, in caso contrario il computer rimane in Standby.

Si inizia con la richiesta dei dati di obbligatoria citazione (a garanzia di regolarità formale del contratto).

Nell'ordine occorre specificare luogo e data di stesura del documento, quindi la destinazione d'uso esclusivo della locazione, per poi indicare lo status giuridico del locatore (privato, ditta o società), unitamente all'Input delle relative complete generalità, che nel caso di privati sono: cognome e nome, codice fiscale, località di residenza, indirizzo e numero dell'indirizzo. Se il locatore è invece una ditta, in luogo di cognome e nome va specificata la denominazione completa; così è anche per le società, e in quest'ultimo



caso vanno anzi aggiunte le precisazioni di numero d'iscrizione nel registro del Tribunale locale e di cognome e nome del legale rappresentante.

Anche per l'altra parte contrattuale, il conduttore (colui che riceve in affitto l'immobile), va ripetuto l'iter di immissione dei dati identificativi: valgono tutte le considerazioni fatte per il locatore. Segue la richiesta di Input relativamente all'immobile oggetto del contratto, e nell'ordine si devono specificare il numero dei locali (stanze, vani), la località d'indirizzo, l'indirizzo, il numero dell'indirizzo, ed eventuali specifiche descrittive (come scala, piano, interno).

Altre indicazioni sono necessarie relativamente alla durata in anni del contratto, alle date d'inizio e fine locazione, all'importo annuale del canone precedentemente calcolato (meglio se con software specifico) e agli oneri accessori di spesa preventivati dal locatore, unitamente alla modalità del relativo pagamento (dettagliando il numero di rate e la scadenza).

Ultima indicazione da fornire è quella relativa al sistema di riscaldamento eventualmente presente (centralizzato oppure autonomo): nel caso di impianto comune, infatti, sul contratto dev'essere inserita una clausola

particolare che vincola il conduttore ad accettare la gestione condominiale del riscaldamento, nonché a rimborsare il locatore delle relative spese per la manutenzione e il consumo. Con Equo Canone 3 la specifica del paragrafo aggiuntivo è automatica, se necessaria.

Durante le fasi di immissione dei dati, un'apposita maschera di input fornisce di volta in volta tutte le informazioni necessarie, affinché l'utente dia risposte adeguate: in azzurro, verde chiaro e giallo viene stampato il testo della richiesta, e più sotto due indicatori di colore giallo delimitano un campo in cui viene poi visualizzato l'inserimento, che può avere un'ampiezza massima di ben 30 caratteri se alfanumerico, oppure di 11 caratteri se puramente numerico. I dati del primo tipo prevedono l'uso dei tasti alfabetici (dalla a alla z), numerici (dallo zero al nove), grammaticali e semigrafici (punti, virgole, segni e barre, molto utili per esempio per scrivere bene gli indirizzi): la visualizzazione su video avviene in colore rosso chiaro, da destra verso sinistra. Per gli input in cifre (per esempio l'importo in lire del canone annuale) sono attivati i soli tasti numerici (dallo zero al nove), per ottenere valori da un minimo di zero (0) a un massimo teorico di novecentonovanta-

Una videata del programma



novemilioni novecentonovanta-novemila e novecentonovantanove (999'999'999) ogni volta. Il numero viene visualizzato in rosso chiaro, già ricomposto a gruppi di tre cifre per una migliore leggibilità; da destra verso sinistra (come per i display delle calcolatrici); è possibile addirittura comporre improbabili valori frazionari premendo il tasto del punto decimale (.).

La conferma per ogni inserimento va data con Return, mentre eventuali correzioni sono sempre possibili, prima della conferma stessa, utilizzando il tasto della freccetta a sinistra (sul Commodore 64 è posto di fianco a quello numerico uno).

Premendo subito e solo il tasto Return, alle richieste di dati numerici, si passa oltre: l'Input sarà allora considerato come inesistente, nullo, e non potrà comparire sul contratto. Per altri tipi di dati (alfanumerici) l'inserimento non può invece avvenire se non si specifica qualcosa.

Per l'immissione di certe indicazioni il multidisplay prevede una sequenza prefissata di opzioni, comodamente esplorabile, "a libro" in avanti e in ritorno, tramite i tasti-funzione F1 e F3; la conferma della scelta operata va poi data con l'usuale Return. Per esempio, quando occorre definire gli status giuridici del locatore e

del conduttore si ha una sequenza di tre opzioni: soggetto privato, ditta, oppure società. Al completamento d'esecuzione della routine di immissione dei dati viene proiettata la videata finale e un messaggio che ricorda all'utente di aspettare affinché la stampante collegata al computer provveda alla stesura del contratto di locazione definito.

È possibile scegliere i fogli singoli in formato standard A4 (da inserire in sequenza), oppure i moduli continui, anche se ovviamente la prima soluzione è quella da preferire; infatti normalmente i contratti sono fogli a più facciate o pagine. Per ottenere la scrittura in fronte/retro basta reinserire nella stampante un foglio appena scritto ed espulso dal rullo, avendo ovviamente cura di capovolgerlo senza girarlo.

La lunghezza di un contratto-tipo, con la spaziatura interlineare prevista (10 millimetri, come d'uso in materia di scritture formali e legali), diventa sempre di almeno tre facciate, ovvero due fogli se si opera in fronte/retro (in questo caso andranno uniti con colla o punti metallici controllando la corretta impaginazione del documento ottenuto).

Normalmente ogni contratto di locazione dev'essere prodotto in tre copie: una per il locatore, una per il conduttore e un'altra per

l'Ufficio del registro. Il programma ne stampa una sola, per cui si può ripetere tre volte la procedura di definizione oppure, si, ricorrere all'aiuto di una fotocopiatrice: la prima pagina di ogni contratto prevede già un'apposita riga (sotto il titolo di apertura in Reverse) in cui specificare a chi è destinata la copia del documento.

Ogni paragrafo della scrittura è evidenziato da un punto nero (oltre che da un numero di riferimento), mentre il testo è stampato in NLQ (Near Letter Quality) per un'ottima leggibilità.

La composizione dei contenuti è automaticamente giustificata ai margini, al fine di conferire all'elaborato un'impaginazione veramente professionale e gradevole all'occhio di chi è tenuto a leggere la lunga sequenza di specifiche.

Il contratto assume validità a ogni effetto quando viene sottoscritto dalle due parti, mediante firma leggibile da apporre nelle apposite righe già previste alla fine dei paragrafi e (ultima pagina) dopo la clausola finale.

Ovviamente locatore e conduttore devono prendere visione del documento, prima di sottoscriverlo e di presentarne una copia all'Ufficio del registro; trattandosi di una scrittura privata può essere cambiato (riscritto) senza difficoltà ogni volta che se ne verifichi la necessità (per esempio per rinnovo contrattuale che preveda varianti).

La stampante di riferimento che garantisce i migliori risultati di impaginazione è la Commodore MPS 1200, altre stampanti possono produrre documenti di diverso Lay-Out per quanto riguarda grafica, caratteri e composizione, anche se ovviamente il contenuto rimane sostanzialmente invariato.

Equo Canone 3 ritorna automaticamente alla videata iniziale di attesa, non appena termina



Un'altra videata del programma.



la routine di stampa dei documenti di volta in volta definiti.

Concludendo, si può affermare che questo programma aiuta molto l'utente, effettuando tutte le procedure più noiose e complesse per la stesura dei contratti di locazione a uso non abitativo.

Gli articoli del Codice civile a cui si fa esplicito riferimento nel testo da sottoscrivere sono i seguenti, di cui si specifica, per corretta informazione delle parti, il contenuto.

• **Articolo 1456** (clausola risolutiva espressa). I contraenti possono convenire espressamente che il contratto si risolva nel caso che una determinata obbligazione non sia adempiuta secondo le modalità stabilite. In questo caso, la risoluzione si verifica di diritto quando la parte interessata dichiara all'altra che intende valersi della clausola risolutiva.

• **Articolo 1584** (diritti del conduttore in caso di riparazioni). Se l'esecuzione delle riparazioni si protrae per oltre un sesto della durata della locazione e, in ogni caso, per oltre 20 giorni, il conduttore ha diritto a una riduzione del corrispettivo, proporzionata all'intera durata delle riparazioni stesse e all'entità del mancato godimento. Indipendentemente dalla sua durata, se l'esecuzione delle riparazioni rende inabitabile quella parte della casa che è necessaria per l'alloggio del conduttore e della sua famiglia, il conduttore può ottenere, secondo le circostanze, lo scioglimento del contratto.

• **Articolo 1609** (piccole riparazioni a carico dell'inquilino). Le riparazioni di piccola manutenzione, che a norma dell'articolo 1576 devono essere eseguite dall'inquilino a sue spese, sono

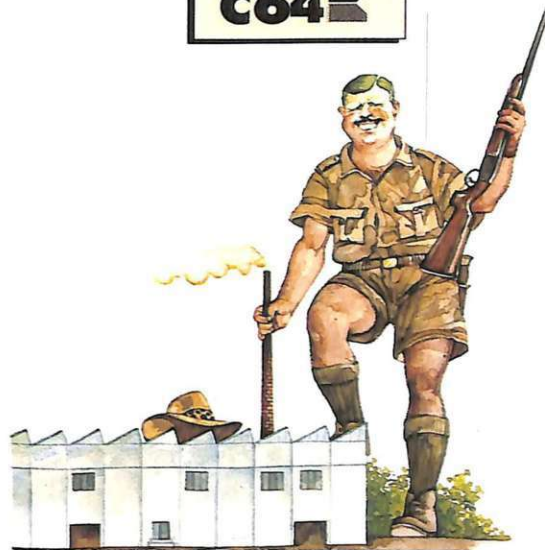
quelle dipendenti dai deterioramenti prodotti dall'uso, e non quelle dipendenti da vetustà o da caso fortuito. Le suddette riparazioni, in mancanza di patto, sono determinate dagli usi locali.

• **Articolo 1756** (rimborso delle spese). Salvo patti o usi contrari, il mediatore ha diritto al rimborso delle spese nei confronti della persona per incarico della quale sono state eseguite anche se l'affare non è stato concluso.

Nel testo contrattuale si fa riferimento anche a un paio di articoli della legge numero 392 del 27 luglio 1978, relativa all'equo canone. Essi sono i seguenti:

• **Articolo 10** (partecipazione del conduttore all'assemblea dei condomini). Il conduttore ha diritto di voto, in luogo del proprietario dell'appartamento locatogli, nelle delibere dell'assemblea condominiale relative alle spese e alle modalità di gestione dei servizi di riscaldamento e di condizionamento d'aria. Egli inoltre ha diritto di intervenire, senza diritto di voto, sulle delibere relative alla modificazione degli altri servizi comuni. La disciplina del primo comma si applica anche qualora si tratti di edificio non in condominio. In tale ipotesi i conduttori si riuniscono in apposita assemblea convocata dal proprietario dell'edificio o da almeno tre conduttori. Si osservano, in quanto applicabili, le disposizioni del Codice civile sull'assemblea dei condomini.

• **Articolo 32** (aggiornamento del canone), sostituito dall'articolo 1 comma 9 *sexies* della legge numero 118, del 5 aprile 1985. Le parti possono convenire che il canone di locazione sia aggiornato annualmente su richiesta del locatore per eventuali variazioni del potere di acquisto della lira. Le variazioni in aumento del ca-



none non possono essere superiori al 75 per cento di quelle, accertate dall'Istat, dell'indice dei prezzi al consumo per le famiglie di operai e impiegati. Le disposizioni del presente articolo si applicano anche sui contratti di locazione stagionale.

Sono di seguito riportati due esempi di definizione contrattuale per locazioni ad uso diverso da quello abitativo, anche al fine di mostrare come sia facile ottenere via software risultati immediati e completi.

• **Il primo caso** tratta di un immobile sito in Bologna, Corso Magenta 114/A, dotato di ben 32 locali; in un edificio al settimo piano, interni 10 e 11. Viene concesso in locazione (ad uso esclusivo di sede di rappresentanza) da un privato a una società, e cioè dalla signora Antonietta De Frivolis, abitante a Milano in Piazza della Resistenza 33/12, codice fiscale DFRNTN24F11L531S, alla Banca Carpigiana S.p.A. (iscritta al Tribunale con numero 9.582.323 e legalmente rappresentata dal signor Giandomenico Vitalone), con sede sociale in Carpi (Mo), Viale delle Mura 450, codice fiscale 03155840360.

Il contratto deve avere una durata di quattro anni, con inizio il



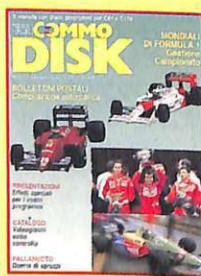
Hai un Commodore 64 o 128?

**Tutti i mesi in edicola
per te, 3 riviste
con tantissimi
giochi e programmi!**



**Radio Elettronica
& Computer**
La rivista
con cassetta
per C64 e C128

Commodisk
Ogni mese
5 programmi
su disco



Tutto Commodore
Ogni mese
un disco
con tanti
programmi

1° gennaio 1990 e termine il 1° gennaio 1994. L'importo annuo del canone è stato stabilito in 130 milioni e 590mila lire, più un preventivo spese accessorie di 17 milioni di lire da corrispondere al locatore tramite 12 rate (mensili) anticipate in scadenza il giorno 10 di ogni mese.

Non avendo l'immobile riscaldamento autonomo, deve comparire in contratto quanto contenuto nel relativo paragrafo di stampa numero sette.

L'accordo viene sottoscritto dalle parti a Bologna il giorno 31 ottobre 1989.

• **Il secondo caso** tratta invece di un magazzino di Napoli, indirizzo Strada Nazionale Treso- na 1875, avente complessivamente quattro locali. Viene dato in locazione (a uso esclusivo di deposito di generi alimentari) da una ditta a un'altra ditta, e cioè dallo Studio Kappa di P. Trenno, con sede in Napoli, Viale Regina Pacis 89/B, codice fiscale PTRTRN56L24F903S, al Ristorante Da Geppo di G. Doro, con sede in Reggio Calabria, Via Scalligera 5, codice fiscale GNNDRO60B01C955R.

Il contratto deve avere una durata di otto anni, con inizio il 1 luglio 1990 e termine il 1 luglio 1998. L'importo annuale del canone è stato stabilito in lire 15 milioni, più un preventivo spese accessorie di lire 450mila lire da corrispondere al locatore tramite due rate (semestrali) anticipate in scadenza il 31 luglio e il 31 gennaio di ciascun anno.

Non avendo l'immobile riscaldamento centralizzato, non deve comparire in documento quanto contenuto nel paragrafo di stampa numero sette del contratto relativo al precedente caso. L'accordo viene sottoscritto dalle parti a Napoli il giorno 15 dicembre 1989.

Daniele Malvasi



Chi non cerca... trova lo stesso!

Da oggi anche gli utenti in possesso del solo registratore a cassette potranno automatizzare la ricerca dei programmi come è possibile fare solo usando un drive. In una manciata di linee Basic abbiamo racchiuso il sistema per costringere il registratore a cercare alla massima velocità i vostri programmi sulla cassetta

Se confrontato con il registratore a cassette, il drive possiede molte allettanti caratteristiche: l'elenco dei programmi contenuti sul disco (directory), il veloce caricamento dei programmi, una buona affidabilità e, ultima ma non meno importante, la possibilità di accedere in modo diretto a tutti i file contenuti nel floppy. Lo sforzo compiuto dal geniale inventore del Turbo Tape ha migliorato notevolmente i tempi di caricamento dei programmi su nastro, ma la ricerca degli stessi sulla cassetta rappresenta sempre la fase più noiosa dell'operazione. Chi possiede un drive, per caricare un qualsiasi programma da disco non fa altro che dare il Loading ed aspettare il caricamento. È infatti la macchina stessa che si occupa di rintracciare sul disco il programma scelto, risparmiando

all'utente ogni fatica. L'utente che si serve del solo registratore a cassette deve invece posizionare il nastro al punto preciso di inizio del programma per caricarlo. Molti utenti sono ricorsi alla classica, ma sempre affidabile, tecnica della carta e della penna. In pratica, sulla cassetta viene posta una spaziosa etichetta su cui si scrivono gli indirizzi (numero di giri del tape counter) di ogni programma. Quando si

vuole caricare un programma dal bel mezzo della cassetta, occorre riavvolgere il nastro, azzerare il tape counter e, con gli occhi fissi sui numerini rotanti, arrestare il nastro all'indirizzo opportuno e procedere al caricamento. L'utility che vi proponiamo questo mese vi libererà dalla noiosa procedura di posizionamento del nastro facendolo per voi.

Come si usa

Ecco, punto per punto, quello che dovete fare per mettere in azione il programma.





Il programma Stratos Format deve essere, per ragioni pratiche, il primo file sulla cassetta. Per questa ragione l'ideale sarebbe che le cassette piene di programmi di cui già disponete, avessero uno spazio vuoto all'inizio del nastro per poter ospitare senza problemi il modulo.

In caso questo non fosse possibile, vi consigliamo di spostare il primo programma in fondo al nastro, posizionandolo come ultimo file, lasciando così uno spazio all'inizio.

Prendete dunque la cassetta che volete formattare e, se non l'avete già fatto, prendete nota dell'indirizzo dei programmi che si trovano su di essa. Ora dovete misurare la velocità di trazione del vostro registratore.

Caricate il programma Speed Format dalla cassetta allegata a questo numero della rivista e lanciatelo.

Premete la barra spazio e selezionate il tipo di cassetta che corrisponde a quella da formattare. Come indicato dal programma stesso, riavvolgete il nastro all'inizio e premete un tasto. A questo punto dovete premere il tasto FFWD sul registratore. Dopo un certo tempo il programma blocca il motorino del registratore e vi esorta a premere il tasto Stop sul

Tavola 1.

Elenco dei comandi del programma Format

F1 : avanti pagina

F2 : stampa lista programmi

F3 : pagina indietro

F4 : inserimento velocità di avvolgimento

F5 : inserimento programmi

F6 : salvataggio del programma

F7 : cambio velocità di avvolgimento

registratore. In seguito vi viene chiesto di inserire il numero di giri che compare sul contatore del registratore. Digitatelo e premete Return. Ancora, premete il tasto FFWD e attendete l'arresto del nastro per ripetere le operazioni appena viste. Questa serie di operazioni viene proposta un numero di volte proporzionale alla lunghezza della cassetta.

Al termine del procedimento il programma vi mostrerà un valore con tre cifre decimali che indica la velocità del vostro particolare registratore. Prendete nota del valore e resettate il computer.

L'operazione successiva è la creazione del modulo Format, da piazzare all'inizio della cassetta da formattare. Caricate il programma Stratos Format dalla cassetta di *Radio Elettronica & Computer* e lanciatelo, come di consueto.

Subito dopo compare un messaggio di input che richiede l'inserimento di un valore il quale indica su quale registratore state operando (vedi oltre). Premete zero seguito da Return.

Cosa importantissima da effettuare a questo punto è informare il programma sulla velocità del registratore in oggetto. Premete F4 e inserite il valore ottenuto precedentemente con il programma Speed Format, e premete Return due volte. Ora dovete assegnare questa velocità a uno dei cinque registratori che è possibile gestire con il sistema Format. Digitate ancora zero e Return. Per vedere l'elenco dei registratori archiviati e le rispettive velocità premete F7. Per conoscere le funzioni degli altri tasti funzione potete osservare la **tavola 1** in queste pagine. L'opportunità di gestire vari registratori vi consente di scambiare intere cassette di programmi già formattate con i vostri amici che vi avranno comu-



Figura 1.
Schermata
principale del
programma
Stratos Speed



nicato le velocità di trazione del loro registratore, oppure di gestire i vostri molteplici registratori (se per qualche arcana ragione ne possedete più di uno).

L'operazione più delicata nella preparazione del modulo Format consiste nell'inserire l'elenco dei programmi sulla cassetta. Premete F5 e digitate il nome del primo programma dopo il modulo Format. Fate attenzione a non superare lo spazio consentito, e individuato dai puntini, per evitare un'impaginazione della tabella errata.

In seguito digitate il valore del contatore del nastro in corrispondenza del programma appena inserito e premete Return. Fate attenzione a inserire i valori in modo che siano sempre di tre cifre (per esempio 002).

Le specifiche, richieste in seguito, istruiscono il programma sulle modalità di caricamento del file inserito (t = turbo, n = normale). L'operazione di inserimento è ciclica e si conclude solo premendo il tasto Return alla richiesta del nome del programma. Ogni volta che vi viene richiesto di inserire il Numero Tape digitate zero, come in questo esempio.

Per consultare l'archivio potete premere i tasti F1 e F3. Per caricare un programma non dovete fare altro che digitare il numero a esso corrispondente sulla pagina correntemente visualizzata.

Una finestra vi illustra i dati di caricamento del programma segnalando l'indirizzo e il tempo di caricamento.

Tutto quello che dovete fare a questo punto, e qui sta il bello, è premere il tasto FFWD sul registratore e aspettare che il programma posizioni il nastro esattamente all'inizio del programma scelto.

Al termine di questa operazione il programma vi suggerisce di premere il tasto Stop sul registratore e di seguito il tasto Play.

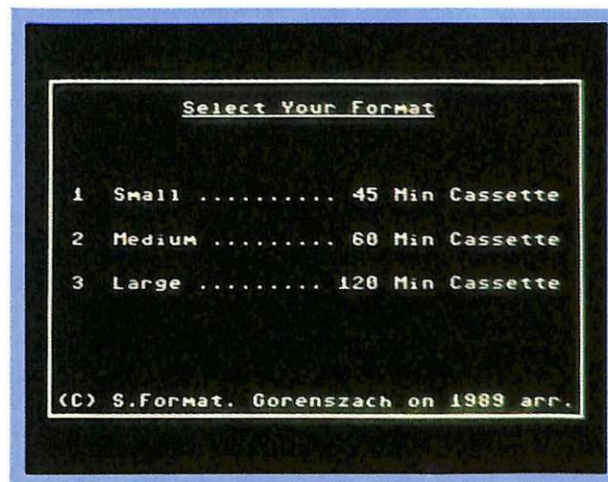


Figura 2.
Schermata
principale del
programma
Stratos Format

Fatto questo dovete solo attendere il caricamento.

Salvataggio del modulo

Il modulo Format può essere salvato su nastro in due modi: da programma (premendo F6 e seguendo le indicazioni) oppure direttamente dal Basic dopo aver arrestato il programma con Run/Stop. Nel primo caso potete salvare il modulo anche in turbo, ma dovete ricordarvi di porre prima di questo il famoso Turbo Tape. Se possedete un programma di Turbo Autostart o Turbo Link (in cui il turbo e il programma costituiscono un file unico) e potrete risparmiare spazio.

Utilizzazione

L'uso delle cassette formattate è molto semplice. Introdurrete il nastro nel registratore, riavvolgetelo all'inizio e infine digitate Load seguito dalla pressione del tasto Return.

Dopo un certo tempo il modulo Format sarà caricato, e voi dovrete ricordarvi di riavvolgere nuovamente il nastro per fissare come punto di partenza per i calcoli del programma Format la posizione iniziale del nastro. Esaminate l'archivio dei programmi su

cassetta e premete il numero corrispondente a quello che vi interessa. Premete FFWD, attendete che il registratore si arresti, premete Stop, premete Play e attendete il caricamento. Alcuni file salvati con sistemi turbo particolari possono dare problemi di caricamento, perché alcune routine del modulo Format interferiscono con essi. Per evitare qualsiasi noia potete resettare il computer dopo la pressione del tasto Stop sul registratore, quando, cioè il nastro è già stato posizionato.

Consigli utili

La fase dell'inserimento dei nomi dei programmi con i relativi indirizzi può causare alcuni problemi. Infatti, se si commette un errore di inserimento, non è possibile effettuare una correzione. Quando si lavora su un computer, però, nulla è impossibile. Con un po' di conoscenza del Basic è facilissimo per chiunque, dopo aver arrestato l'esecuzione con Run/Stop, listare le ultime linee Data del programma, individuare i dati errati e modificarli secondo le necessità.

Un unico accorgimento: rispettare la sequenzialità dei dati, le relative variabili con indice e la struttura dei dati (utilissima è



UTILITY



l'osservazione dei dati già inseriti). Con un minimo di pratica può risultare più facile inserire i dati direttamente nel listato piuttosto che farlo dal programma.

La capacità di immagazzinamento dati del programma Format è molto grande. Questo fattore permette, al limite, di inserire in un unico modulo, salvato

su una cassetta dedicata esclusivamente ad esso, gli indirizzi di tutti i programmi di tutte le cassette che avete. Il problema è costituito solo dalla scarsità di spazio per definire su quale cassetta si trova il tal programma al tale indirizzo. La soluzione che ci appare più conveniente è di inserire tale informazione nello stesso spazio del nome. Una volta generato e salvato il modulo dovete solo caricarlo, lanciarlo, scegliere fra la miriade di programmi dell'archivio quello da caricare, prendere la cassetta su cui si trova, inserirla nel registratore, riavvolgerla e premere il tasto corrispondente al programma scelto.

Raffaele Zanini
Maurizio Gorenszsch

ELETRONICA INFORMATICA COMUNICAZIONE

Informazione
specializzata
e completa

Gruppo Editoriale

JCE

Via Ferri, 6
20092 CINISELLO
BALSAMO (MI)

SMAU
1989

FUTURE

N° 7/8 LUGLIO - AGOSTO 1989 - L. 6.000

COMPUTER

ANNO IV - N° 7/8 - Luglio/Agosto 1989

EDIZIONE



Gruppo Editoriale
JCE

Allo SMAU
**CERCA LE RAGAZZE JCE
& VINCI**



Gruppo Editoriale
JCE

L'Informazione Qualificata

SE HAI PERSO UN NUMERO... ...HAI PERSO UN TESORO

Come fai se l'arretrato non ce l'hai?

Ti sei perso un numero, o addirittura più numeri, di RadioElettronica&Computer? In queste pagine ti viene offerta l'opportunità di rimetterti in pari. Di ogni arretrato troverai l'elenco dei progetti pubblicati quel mese. Affrettati a spedire la richiesta utilizzando il coupon pubblicato in queste pagine: riceverai subito a casa tua il numero o i numeri che ti interessano senza aggravio di spese postali.

1 - Gennaio 1988 - L. 10.000 - Prosegue il corso di scacchi. Agocom: il software che fa l'agopuntura. Anche dati: migliaia d'informazioni utili in tempo reale. Smart modem 21-22. Con Super Basic avrete 42 comandi per la gestione del drive e della grafica. Gioco: Paura nella torre. Basic Lightning: poligoni regolari, spezzate chiuse e aperte e i comandi per il movimento degli sprite. Recensione: Street baseball. Gioco: Caccia all'obelisco. Fai da te: Comgun, bersaglio mobile. Magic Window: sempre più magica. Cruciverba: un programma creatore di schemi. Tips & Tricks: nuova rubrica dedicata ai lettori molto pratici con consigli di programmazione. Utility: una mini espansione per gli sprite.

2 - Febbraio 1988 - L. 10.000 - Comgun: software su cassetta per il tiro a segno presentato sullo scorso numero. Scacchi: tattiche antiche. Cruciverba: solutore elettronico. Circuiti con l'hard copy. Recensione: Combat school, simulatore di addestramento. Stampante novità della Citizen. Gioco: chi salverà la galassia? Archivio videocassette. Floracom: il computer pensa alle vostre piante. Gamebasic: per animare il gioco. Protector, la directory che rende illeggibile il contenuto dei vostri dischetti. PetSpeed e Magic Window presenti sul vostro Commodore.

3 - Marzo 1988 - L. 10.000 - Scacchi: l'attacco di minoranza. Calcolo enigmatico. Anche dati: a Wall Street con The Source. Iconbase permette una facile archiviazione dei vostri dati. Recensione: è di turno il basket. Magic Window sempre più magica. Gioco: attenti agli animali robot. Gioco: forza uomo gatto. Fai da te: tutti atleti con Supergym. Disk Basic aggiunge ben 34 comandi al Basic per padroneggiare il drive. Il software professionale di Floracom. Didattica: ecco il primo appuntamento dedicato alla geometria analitica. Senza rimpianti il passaggio da Commodore a Amiga.

4 - Aprile 1988 - L. 10.000 - Basic Lightning: concludiamo l'analisi degli attributi. Wizard Key porta a 16 i tasti funzione. Eserciziario di trigonometria. Scacchi: un finale tra Re e pedone senza che venga giocato. Corsa spaziale in 3 dimensioni. Tanta RAM in una semplice cartidge. Recensione: Un gioco proprio come Platoon. Gioco: gli alieni invadono Urano. Segreteria telefonica intelligente. 4 routine per gli effetti speciali. Missione Gold Basic.

5 - Maggio/Giugno 1988 - L. 10.000 - Potete creare introduzioni grafiche degne dei più noti videogame. Phonextra: il software su cassetta della vostra segreteria telefonica. Interfaccia Amiga-like per il vostro C64. Music Basic: la più completa espansione di gestione del Sid. Gioco: 3 gorilla in città. Recensione: olimpiadi dagli effetti esilaranti. A tutta musica con il C64. L'espansione RAM migliora. Fai da te: la macchina della verità. Gioco: distruggete l'imperatore del sistema Delta. Nastroteca su disco. Scacchi: ultima puntata.

6 - Luglio/Agosto 1988 - L. 10.000 - Con Logic Val valutate complesse espressioni matematiche. Tecniche di caricamento di loader e turbo. Un disegnatore per i soft-ware sprite. BMX: una gara di biciclette. Gioco: Giana delle meraviglie. Digitalizzatore in real-time. Seconda puntata del corso di musica: scale musicali per pianoforte e la programmazione del Sid. Un'utility che vi propone una griglia su cui disegnare. Drive basic, una super espansione. Ultima puntata del corso di scacchi. Recensione: Firefly, salvate la Terra dall'invasione nemica. Software su cassetta della macchina della verità. Una routine per scovare gli errori nei dischi. Prima parte degli esercizi sullo studio delle funzioni.

7 - Settembre 1988 - L. 10.000 - Basic test: 4 programmi per migliorare la conoscenza del basic. Dan Dare: un'avventura mozzafiato. Gli errori del computer. Fuga da Zart: il più irresistibile tra gli arcade. Le novità sul drive 1541 II. L'eserciziario di matematica: le funzioni trascendenti. Realizzare animazioni e giochi con Animation C.K. 1a puntata sullo standard MIDI. Grafik 2000: gestione avanzata dell'alta risoluzione. L'hardware del C64. Gestioni campionati. Recensione: Bedlam, uno spaziale con ben 16 schermi diversi.

8 - Ottobre 1988 - L. 10.000 - Action Replay e The Expert Trilogy: due cartucce dalle prestazioni eccezionali. Guerra in Vietnam: i dodici della squadriglia blu. Quattro caricatori per risolvere i problemi di duplicazione e trasferimento programmi da nastro a disco. Flying Shark II: una squadriglia di caccia nei cieli d'Africa. Ancora trucchi e scorciatoie per il tuo C64. Recensione: Il perfido Elvin vuole distruggere il mondo. Shoot'em up: un programma per realizzare videogames. Sprite & Graphic Basic: un'espansione per la gestione degli sprite e dell'alta risoluzione. L'architettura del bus: il microprocessore. Gli "errori" del computer. Velox: un colloquio contemporaneo e diretto di ben sette persone con il computer. Come gestire al massimo le possibilità di Mastercard.

9 - Novembre 1988 - L. 10.000 - Più compatte le schermate del Koala. Il linguaggio dello standard Midi. Gioco: i misteri di Netherworld. Massima qualità con Sound 64. Roby Robot: l'hardware che pilota gli elettrodomestici. L'angolo di Amiga. Shoot'em up: gli sprite. Recensione: Target Renegade. Seconda puntata di SG Basic. Come gestire Velox.

10 - Dicembre 88/Gennaio 1989 - L. 10.000 - Utility: 99 font per accessoriare i propri programmi. Midi: ecco come usare i messaggi. Fai da te: accendi col computer il tuo albero di Natale. L'intramontabile Mickey Mouse. SG Basic: la stesura dei programmi. Teledisco nel C64. Shoot'em up: scenario ed effetti sonori. Computer Graphics con Amiga. Didattica: calcolo integrale di una funzione variabile. Arrow of death: l'avventura più bella.

1 - Febbraio 1989 - L. 10.000 - Avventura: i segreti di Arrow of Death. Tutti i segreti per fare word processing. Listati: l'utility per esporti e commentarli. Recensione: a scuola di guerra. Roby Robot, seconda puntata. Pagelox: la cartuccia per stampare. Fai da te: il joystick che obbedisce alla tua voce. Amiga: alla scoperta dell'hard-ware. Shoot'em up: come sistemare i protagonisti. Ultima puntata di SG Basic.

2 - Marzo 1989 - L. 10.000 - Easy Script: ultima puntata. L'hardware per la gestione dell'acquario. Sprite giganti Hi-Res. Text to text: per abbattere le barriere di incompatibilità fra Ascii, PetAscii e codici video. Il gioco dell'androide volante. Il software per far eseguire al joystick i comandi impartiti a voce. I linguaggi di Amiga.

3 - Aprile 1989 - L. 10.000 - Interfaccia Midi: Come collegare strumenti, computer, pedali e suonare in armonia. Software: Suddividere in livelli lo scenario del gioco. Gioco: Il canguro d'acciaio. Utility: Il mondo dei suoni digitalizzati. Accessori: Una scheda per fissare su Rom tutto il software. Fai da te: Elettronica al servizio dello sport. Amiga: Multitasking permette di eseguire contemporaneamente più programmi.

4 - Maggio 1989 - L. 10.000 - Standard Midi: Effetti speciali professionali. Didattica: Un corso per insegnare il linguaggio Logo. Utility: Programma che permette di effettuare compattazioni di file. Accessori: Straordinarie capacità musicali di Amiga. Fai da te: Misuratore di rilassamento psico-fisico. Gioco: Simulazione di Guerra. Amiga: Come lavorare alcune delle principali funzioni nella libreria Exec.

5 - Giugno 1989 - L. 10.000 - Con gli accessori per Amiga, il computer vede e fa vedere - Gioco: West Bank, il game dal grilletto facile - Midi: Arpeggi con joystick e pedali - Utility: HI-RES: Convenzione di schermate - Una macchina di Byte d'oro - Didattica: Corso per conoscere il computer con il Logo - Espansioni: Struct Basic istruzioni per la manipolazione degli sprite - Fai da te: Dispositivo elettronico che rileva il battito cardiaco.

6 - Luglio/Agosto - L. 10.000 - La stampante del momento: Star LC10 - Ultra Basic: 200 Programmi in più - Le basi per sviluppare in Logo - Una cartella clinica elettronica - Gioco: Bouncer, professione Pallina - I reconditi perché di Amiga Dos - Cinque Utility in linguaggio macchina per C64 - Musiche per ogni programma - Contabilità: Un minuto per districare la matassa - Shell QPL: L'interprete di comandi - Un buon esempio per imparare.

7 - Settembre - L. 10.000 - Utility: Message Maser e Picture Start: Videomania alla ribalta - Accessori: Action Replay MK5 - Logo: ricorsività, grafica, calcoli e numeri - Suoni: dimmi di Sid! - Exec: Librerie e dispositivi - Equo Canone 2: contratto automatico a norma di legge.

Tagliando richiesta arretrati

Per ricevere a casa, senza aggravio di spese postali, l'arretrato o gli arretrati che ti interessano, compila e spedisce subito questo tagliando in busta chiusa a:

**RadioElettronica&Computer
Gruppo Editoriale JCE
via Ferri 6 - 20092 Cinisello Balsamo (MI)**

Sì! Inviatemi i seguenti numeri arretrati di RadioElettronica&Computer:

me/mese di

Cognome Nome

Via

Cap Città Prov

☐ Allego L.

☐ Allego ricevuta di versamento di L. sul
conto corrente postale n. 351205 intestato a Gruppo Editoriale
JCE - via Ferri 6 - 20092 Cinisello Balsamo (MI)

☐ Allego assegno di L. non trasferibile
intestato a Gruppo Editoriale JCE

Data Firma

Una spremuta di Sprite

Prima puntata

Iniziamo questa attesissima rubrica analizzando la struttura di un programma che gestisce in modo decisamente avanzato gli sprite hardware del C64, spremendo e usando al limite tutte le potenzialità nascoste del Vic II e della arcana struttura di funzionamento del Raster. Sul prossimo numero vedremo, in modo più approfondito, come sfruttarne le funzioni

Raster è un termine molto usato nell'ultimo anno, e lo dimostra l'enorme mole di lavoro e documentazione apparsa su tutte le riviste. Sicuramente l'aspetto teorico è più che noto, ma purtroppo scarseggiano programmi che sfruttino in modo innovativo e utile questa peculiarità del C64.

L'obiettivo che ci siamo proposti è di usare le conoscenze sul raster, e in generale sul Vic II, per ottenere qualcosa di analogo a una finestra grafica. L'idea è stata di creare una finestra hardware, cioè indipendente dagli eventi esterni, in modo che il pro-





Listato 1.

Esempio di routine per la gestione del Raster

```
02A7 78          sei
02A8 A9 CB      lda #SCB
02AA 8D 14 03   sta $0314
02AD A9 02      lda #S02
02AF 8D 15 03   sta $0315
02B2 AD 11 D0   lda $D011
02B5 29 7F      and #S7F
02B7 8D 11 D0   sta $D011
02BA A9 64      lda #S64
02BC 8D 12 D0   sta $D012
02BF A9 01      lda #S01
02C1 8D 1A D0   sta $D01A
02C4 A9 7F      lda #S7F
02C6 8D 0D DC   sta $DC0D
02C9 58         cli
02CA 60         rts
02CB AD 19 D0   lda $D019
02CE 8D 19 D0   sta $D019
02D1 A2 07      ldx #S07
02D3 A0 FB      ldy #SFB
02D5 A5 02      lda $02
02D7 F0 04      beq $02DD
02D9 A2 00      ldx #S00
02DB A0 64      ldy #S64
02DD 86 02      stx $02
02DF 8E 20 D0   stx $D020
02E2 8C 12 D0   sty $D012
02E5 4C 31 EA   JMP SEA31
```

SYS 679 per dividere in due lo schermo.

gramma non dovesse controllare ciò che succedeva sullo schermo. Per non illudere nessuno spieghiamo subito cosa intendiamo per indipendenza.

Uno sprite, quando è posto sullo schermo, vi rimane, qualunque cosa si disegni in seguito sullo stesso schermo.

In pratica è come se lo sprite fosse un altro schermo a sé stante, oppure posto su un piano diverso.

Questo è possibile perché gli sprite sono gestiti dal Vic II (coprocessore grafico), che esegue quasi tutte le operazioni riguardanti gli sprite e la grafica in genere, quando il 6510 (microprocessore del C64) non fa nulla, cioè durante la fase uno del clock di sistema (clock basso).

La finestra che proponiamo, l'avrete già capito, si ottiene appaiando tra loro alcuni sprite.

Certo che con soli otto sprite c'è poco da stare allegri, ma per fortuna esiste il raster, e programmando degli opportuni interrupt è possibile ripetere tante righe di

sprite una di seguito all'altra, ottenendo una matrice di pixel abbastanza grande.

Ogni microprocessore, anche quello del C64, a intervalli di tempo più o meno regolari blocca l'esecuzione del programma principale e fa avvenire ciò che si chiama interrupt. Il microprocessore 6510 viene costretto a saltare indirettamente alla locazione puntata da \$0314-\$0315 [JMP(\$0314)], e quando incontra l'istruzione Rti riprende i processi che stava eseguendo prima della richiesta dell'interrupt. Nel C64, non solo il 6510,

ma anche il coprocessore grafico (Vic II) è in grado di provocare degli interrupt.

Questi permettono di controllare certe caratteristiche della grafica, e si abilitano settando gli opportuni bit della locazione \$D01A, con i significati riportati in **tavola 1**.

Il bit più interessante è il bit zero. Questo permette di far avvenire un interrupt ogni volta che il pennello elettronico ha raggiunto una certa riga dello schermo. La riga si sceglie scrivendo il valore voluto nella locazione \$D012, che contiene gli otto bit meno significativi della posizione del raster.

Il nono bit della posizione, o Msb (Most Significant Bit), è il bit sette della locazione \$D011 e serve per esprimere valori superiori a 255.

Listato 2.

Routine utente per cambiare il colore di ogni linea di sprite

```
$CDDC      ldy $CD39
           lda $CDF8,y   Contiene i colori per ogni riga di sprite
           sta $D027
           sta $D028
           sta $D029
           sta $D02A
           sta $D02B
           sta $D02C
           jmp $C238
$CDF8 : 00 07 04 02 01 0E 0F 08
```

Il segreto

La tecnica per ottenere un interrupt raster è piuttosto semplice e abbastanza conosciuta dai nostri più affezionati lettori di questa rivista.

Si può riassumere in questo modo:

- 1) Disattivare gli interrupt con sei.
- 2) Modificare il vettore di interrupt affinché punti alla nostra routine di gestione e controllo interrupt (locazioni vettore \$0314-\$0315).



- 3) Scrivere in \$D012 il valore di linea video (ogni linea è alta un pixel) a cui deve avvenire la richiesta di interrupt.
- 4) Porre a uno il bit zero della locazione \$D01A (lda #\$01: sta \$D01A) e riattivare gli interrupt (CLI). Il punto 4) attiva l'interrupt raster. Un buon consiglio è di disattivare ogni tipo di interrupt tranne quello raster (lda#\$ - 7F:sta\$ DC0D).
- 5) La routine di gestione degli interrupt avrà allora il compito di sostituirsi alla vecchia routine di sistema (\$EA31). Per semplicità, la nostra routine chiamerà una volta per quadro la routine \$EA31.
- 6) La prima cosa da fare quando si verifica un interrupt è di porre a uno la seguente locazione \$D019, altrimenti non avverranno altri interrupt (lda \$D019:sta \$D019 mette a uno il bit uno se è avvenuto un interrupt).

Ovviamente questi compiti devono essere svolti da una routine in linguaggio macchina. Si veda l'esempio del **listato 1**, oppure si studi la routine del programma con inizio a \$C0FF.

Ricapitolando, disegniamo sei sprite appaiati (144 pixel in orizzontale e 21 pixel in verticale), poi dopo 16 righe di schermo (16 pixel) provochiamo un interrupt

raster, per mezzo del quale incrementiamo di 21 pixel l'ordinata dell'intera riga di sprite e allo stesso tempo modifichiamo i puntatori alle animazioni. Così facendo, la finestra si può allungare quanto si vuole in verticale, fino a occupare tutto lo schermo visibile.

Il tutto sembra semplice, ma senza una corretta conoscenza della teoria dell'operazione non c'è verso di farlo funzionare.

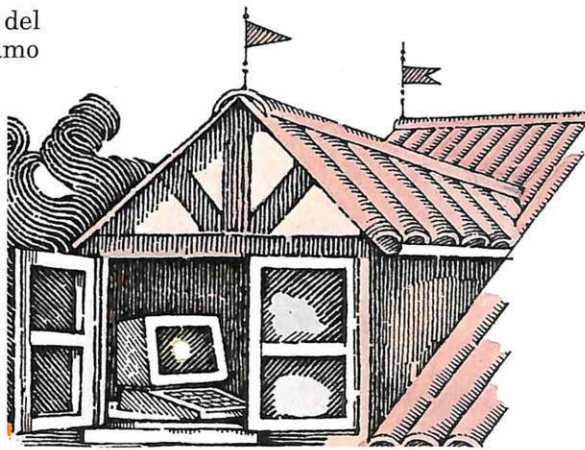
Il Vic II all'inizio di ogni riga di schermo confronta l'ordinata degli sprite con la posizione del pennello elettronico.

Nel caso che le due posizioni coincidano, il Vic II preleva il puntatore all'animazione dello sprite in questione e incrementa un contatore interno a sei bit, in modo da leggere tutti i 63 dati che definiscono l'animazione.

In pratica il Vic II ricopia 3 byte di ogni sprite (24 bit) ogni linea di quadro.

Notare che durante questo periodo il 6510 rimane parzialmente fermo, ragion per cui gli sprite rallentano l'esecuzione dei programmi.

Se si usa una finestra di dimensioni normali, poiché vengono usati circa 35 sprite si può



avere un rallentamento del programma principale di un buon 20 per cento.

Sincronizzazione

Ora risulterà chiara la definizione di finestra hardware. Essa è gestita dal Vic II, lasciando solo pochi compiti al 6510, per esempio dopo che è stata disegnata per intero la prima riga di sprite e bisogna cambiare i valori dei puntatori alle animazioni. Questa operazione va fatta il più velocemente possibile, altrimenti compaiono strane sovrapposizioni e tremolii.

Per ottenere il massimo sincronismo il programma procede in questo modo: provoca un interrupt raster di 4 linee prima che sia raggiunta la posizione della riga seguente di sprite. Aggiorna la posizione y degli sprite, e testa la locazione \$D012 (posizione del pennello elettronico) fino a che questa non eguagli la posizione degli sprite.

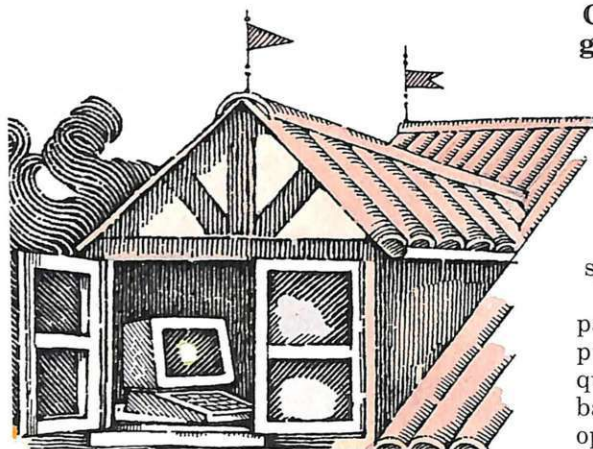
```
lda # riga inizio dello sprite
cmp $D012 ; pos del pennello
        elettronico
bne #$FD
```

A questo punto i puntatori alle animazioni vengono modificati. Ricapitolando, è impossibile ag-

Tavola 1.

Significato dei bit della locazione \$D01A

bit0:IRQ di comparazione quadro (raster)
 bit1:IRQ di contatto sprite con fondo
 bit2:IRQ di contatto sprite con sprite
 bit3:IRQ triggerato per penna ottica



giornare valori di più di sei sprite alla volta, per la mancanza di cicli macchina tra una riga raster e l'altra.

La finestra funziona bene qualunque siano le sue dimensioni. Le condizioni limite possono essere raggiunte se si usano sei sprite affiancati e più di cinque sprite in verticale.

In questo caso perché non si abbiano tremolii è necessario che la finestra sia in determinate posizioni. (la migliore è $X=n*8$ e $Y=n*8+3$ dove $n=1,2,3,\dots$).

Vale anche la tabella riportata in **tavola 2**.

Considerazioni generali

Il programma usa sempre gli sprite tra zero e cinque, dunque restano liberi solo altri due sprite, e precisamente lo sprite sei e sette.

La memoria occupata dalla window può essere anche quella sotto la rom basic (\$A000 \$C000), oppure quella sotto la rom kernal (\$E000 \$FFFF). Il tutto è trasparente all'operatore.

A proposito di memoria, ricordo che le locazioni comprese tra (\$1000 e \$2000), tra (\$9000 e \$A000), tra (\$D000 e \$E000) e logicamente tutta la pagina zero e uno tra (\$0000 e \$0400) non sono accessibili alla memoria sprite, perché già occupate dal generatore caratteri. Inoltre il programma occupa tutta la memoria compresa tra \$C000 e \$CDDF. Se fosse necessaria questa area di memoria basterà rilocare il programma con Zoom, oppure modificare l'indirizzo di base nel programma in assembler.

Vengono inoltre usate le seguenti locazioni di pagina zero: \$02,\$FB,\$FC,\$FD,\$FE.

Nota: Quando una finestra è attivata, le comunicazioni tra il computer e le altre periferiche possono non funzionare, pertanto prima di usare il bus di comunicazione è meglio spegnere la finestra (sys 49152,10:0).

Per gli esperti

Anticipiamo ora qualche informazione sulla routine centrale del programma per i più esperti che desiderassero effettuare qualche aggiunta.

Come vedremo meglio nella terza puntata, il programma, dopo avere modificato i puntatori alle animazioni salta a una routine che può essere definita dall'utente. In questo modo si può, per esempio, modificare il colore degli sprite ogni 21 pixel, oppure cambiare i colori del multicolor e così via. La routine deve cominciare a \$CDDC, non modificare assolutamente il registro X del 6510 e, completato il suo compito, saltare con un JMP a \$C238.

Creando una propria routine, ed usando sei sprite in orizzontale e più di cinque in verticale, possono comparire evidenti sfarfallamenti. Per evitare ciò bisogna trovare i valori che pokati nella tabella Tabs1 (\$CDA6) fanno sparire questo fenomeno. Normalmente i valori vanno da due a 30. Per eseguire questa procedura bisogna attivare la finestra, in caso di sfarfallamenti cambiare i valori in Tabs1 e richiamare la Routine Uno. Nota Tabs1+5 contiene il valore di sincronismo se ci sono sei sprite in verticale, Tabs1+6 per sette sprite in verticale e così via. Provate la routine del **listato 2** che cambia i colori ogni 21 pixel.

Nicola Chiminelli
(continua)

Tavola 2.

**X e Y sono le coordinate di posizionamento della finestra.
Il parametro n è uguale a (0,1,3,4...).**

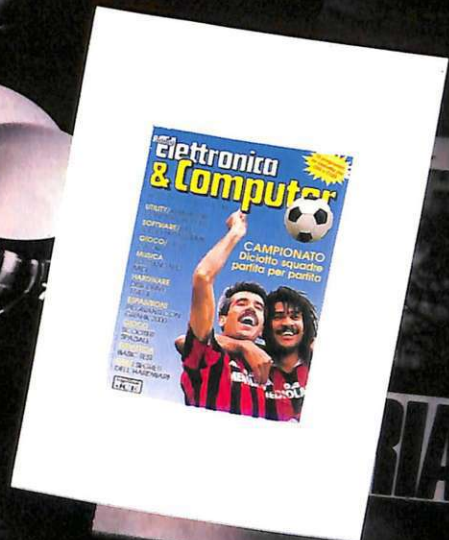
Posizioni migliori:

$X = n*8$
 $X = n*8+1$
 $X = n*8+4$
 $Y = n*8$
 $Y = n*8+3$
 $Y = n*8+5$

Posizioni sconsigliate:

$X = n*8+2$
 $X = n*8+6$
 $Y = n*8+6$

OBIETTIVO CHIAREZZA



La testata che state leggendo
è iscritta al Consorzio e come
tale è soggetta agli obblighi
previsti dal regolamento.



Consorzio Stampa Specializzata Tecnica

Libera associazione di editori nata con l'obiettivo
di tutelare il settore delle riviste specializzate e tecniche
e di fornire dati certificati nonché
informazioni chiare sulla loro diffusione e readership...
per la più facile e sicura pianificazione pubblicitaria.

SUPERSPORT

Consorzio Stampa Specializzata Tecnica

Via Pantano, 9
20121 MILANO
02/2298/235

PROVE HW:
ATARI PC E
DRIVE CUMANA

ANATOM
DI UN



Grafica in bassa risoluzione e drive

Ecco chi sono i protagonisti di questa puntata dedicata alla mega espansione Ultra Basic. Creare finestre e implementare la tecnica degli overlay non sarà più un problema: poche e semplici istruzioni faranno per voi tutto il lavoro necessario

Gli argomenti di questa puntata di Ultra Basic sono la grafica in bassa risoluzione, o modo testo, e la gestione del drive. In entrambi i casi l'espansione si dimostra efficacissima semplificando notevolmente la vita del programmatore.

Combinando opportunamente alcune semplici istruzioni vi sarà possibile, per esempio, realizzare finestre, scrollare porzioni dello schermo nelle quattro direzioni e implementare la tecnica degli overlay. Come di consueto divideremo le istruzioni per ambito di applicazione.

Grafica in bassa risoluzione

- **Color:** fissa il colore di linea, del bordo e del fondo dello schermo. La sintassi di questa istruzione, che può essere usata sia in modo diretto sia in modo programma, è: color l, b, f.

Il primo parametro rappresenta il colore di linea, il secondo il colore del bordo dello schermo, e infine il terzo il colore del fondo del video.

Per esempio con Color 0, 1, 3 il colore di linea diviene il nero, mentre il bordo e il fondo dello

schermo vengono colorati rispettivamente di bianco e viola.

- **Fcol:** riempie una porzione del video con un determinato colore. Questa istruzione, che come la precedente può essere usata sia in modo programma sia in modo diretto, agisce esclusivamente sulla memoria del colore, e quindi non produce alcun effetto se sullo schermo non c'è nulla. La sintassi dell'istruzione è: fcol x, y, lx, ly, cl. I primi due parametri rappresentano le coordinate dell'angolo superiore sinistro della porzione rettangolare da colorare. I due parametri successivi rappresentano rispettivamente la larghezza e l'altezza della porzione. L'ultimo parametro è il colore. Per esempio per portare in nero l'intero contenuto dello schermo basta usare l'istruzione: fcol 0,0,40,25,0. I primi due parametri devono essere compresi rispettivamente fra 0 e 39 e fra 0 e 24. I parametri lx e ly devono invece essere compresi fra 1 e 40 e fra 1 e 25; inoltre deve essere: $x+lx < 41$ e $y+ly < 26$. L'ultimo parametro deve essere ovviamente un intero fra 0 e 15 poiché rappresenta il codice di un colore.

Listato 1

```
1 REM -----
2 REM -
3 REM - SUP DEMO -
4 REM -
5 REM -----
6:
10 REPEAT
20 SUP 0,0,20,12,1
30 SUP 20,13,20,12,1
40 GET AS
50 UNTIL AS <>"
60 FOR I= 1 TO 25
70 SUP 0,0,40,25,0
80 GETKEY AS
90 NEXT
```

*Listato 1.
Poche linee
Basic per un
effetto grafico
veramente
notevole.
Provare
per credere*



- **Fchr:** questa istruzione è analoga alla precedente con la sola differenza che permette di riempire una porzione dello schermo, non con un determinato colore ma con un carattere. Quindi questa istruzione, a differenza della precedente, agisce sulla memoria video.

La sintassi dell'istruzione è: fchr x, y, lx, ly, cr. Il significato dei primi quattro parametri è identico a quello degli omonimi parametri dell'istruzione precedente. Il parametro cr rappresenta il codice video del carattere con cui si vuole riempire la porzione di schermo selezionata. Per esempio, per riempire le prime due linee dello schermo con la lettera a si dovrà usare la seguente istruzione: fchr 0,0,40,2,1.

- **Fill:** questa istruzione equivale praticamente alla somma delle due istruzioni precedenti. Permette infatti, di riempire una determinata regione rettangolare dello schermo con un carattere in un certo colore. Questa istruzione quindi agisce sia sulla memoria video sia su quella del colore. La sintassi dell'istruzione è: fill x, y, lx, ly, cr, cl. Il significato dei parametri che compaiono nell'istruzione è quello descritto nelle istruzioni precedenti. Per esempio, l'istruzione fill 0,0,40,4,81,4 equivale alle istruzioni fcol 0,0,40,4,4 e fchr 0,0,40,4,81.

- **Scrinv:** inverte (cioè pone in Reverse ciò che è visualizzato in modo normale e viceversa) il contenuto di una porzione rettangolare dello schermo. La sintassi dell'istruzione è: scrinv x, y, lx, ly. Il significato dei parametri che compaiono nell'istruzione è quello solito.

- **Scopy:** copia una porzione rettangolare dello schermo. La sintassi dell'istruzione è: scopy x, y, lx, ly, dx, dy. Il signifi-

Listato 2

```

1 REM _____
2 REM -
3 REM - FLASH DEMO -
4 REM -
5 REM _____
6:
8 PRINT" (CLEAR)"
10 COLOR 0,15,15
20 FILL 0,0,40,10,32,15
30 LOCATE 10,9,9,"- FLASH-1 PROVA -"
40 FLASH 0,1
50 PAUSE 3
60 COLOR 1,15,15
70 FILL 0,10,40,15,32,15
80 LOCATE 10,18,"- FLASH-2 PROVA -"
90 FLASH 1,10
100 PAUSE 3
110 FLASH
    
```

*Listato 2.
Esempio d'uso
dell'istruzione Flash.
Questa istruzione
è molto utile
per realizzare scritte
che richiamano
l'attenzione*

cato dei primi quattro parametri è quello solito, mentre gli ultimi due parametri permettono di specificare di quanto viene spostata nelle due direzioni la regione selezionata. Per esempio l'istruzione :scopy 0,0,40,2,0,10 permette di spostare verso il basso di 10 linee le prime due righe dello schermo. Ricordate che dal momento che dx e dy rappresentano uno spostamento relativo dovranno essere verificate le relazioni: $lx+dx<41$ e $ly+dy<26$.

- **Change:** scambia il contenuto dei due schermi messi a disposizione da Ultra Basic. Il primo dei due schermi è memorizzato a partire da \$0400 e la memoria del colore a esso associato parte dall'indirizzo \$d800. Il secondo schermo è memorizzato a partire dall'indirizzo \$f000 e la memoria del colore corrispondente parte dall'indirizzo \$f400. Lo schermo di default è quello memorizzato in \$0400.

- **Sup:** scolla verso l'alto una porzione rettangolare dello schermo. La sintassi dell'istruzione è: sup x, y, lx, ly, m. I primi quattro parametri hanno il solito significato. Il quinto parametro permette di controllare la moda-

lità dello scrolling. Se maiuscolo (m=1) lo scrolling è con wrap around, mentre se minuscolo (m=0) lo scrolling è senza wrap around. La **listato 1** contiene un breve programma d'esempio che mostra ciò che si può fare con questa istruzione.

Prima di lanciare il demo, fate in modo che sullo schermo sia visualizzato qualcosa.

- **Sdown:** scolla verso il basso una porzione rettangolare dello schermo. La sintassi dell'istruzione è identica a quella dell'istruzione precedente.

- **Sright:** scolla verso destra una porzione rettangolare dello schermo. La sintassi è identica a quella dell'istruzione Sup.

- **Sleft:** scolla verso sinistra una porzione rettangolare dello schermo. La sintassi dell'istruzione è identica a quella dell'istruzione Sup.

- **Flash:** questa istruzione permette di realizzare un gradevole effetto grafico con pochissimo sforzo. Con Flash potete far lampeggiare a una velocità prefissata tutti i caratteri visualizzati in un determinato colore.



La sintassi dell'istruzione è: flash c, v. Il parametro c rappresenta il colore dei caratteri da far lampeggiare, mentre il parametro v misura la velocità del lampeggio in 60esimi di secondo (v=1 velocità massima, v=255 velocità minima). La **figura 1** contiene un esempio d'applicazione di questa istruzione. Il **listato 3** contiene il listato di un demo in cui compare la maggiorparte delle istruzioni grafiche di questa puntata. Lanciatelo perché ne vale veramente la pena.

La **figura 2** contiene una schermata tratta dal demo. La schermata è stata prodotta con pochissime linee di programma.

Gestione del drive

- **Dload:** equivale all'istruzione Load ma a differenza di questa non necessita del ,8 finale.

La sintassi dell'istruzione è: dload"nomefile" dove nomefile è il nome del file che si vuole caricare (in pratica Dload è una abbreviazione di Load).

Tenete presente che Dload carica il programma a partire dall'inizio dell'area Basic. Per effettuare il caricamento nella zona d'origine (Load, 8, 1 in Basic standard) dovete usare l'istruzione dload"nomefile",1.

- **Dsave:** è l'equivalente di Dload per l'istruzione Save.

Anche in questo caso l'unica differenza con l'istruzione del Basic standard sta nel risparmio dello otto finale.

La sintassi dell'istruzione è: dsave"nomefile" dove nomefile è il nome da assegnare al file da salvare.

- **Dmerge:** carica un programma in memoria e lo accoda a quello eventualmente già presente. La sintassi di questa istruzione è identica a quella dell'istruzione Dload. È da sottolineare il fatto che spetta al programmatore garantire la compatibilità dei numeri di linea dei programmi da fondere.

- **Chain:** permette di caricare e lanciare un programma Basic. La sintassi è identica a quella dell'istruzione Dload.

Con questa istruzione si può, finalmente, evitare di ricorrere al trucco del buffer di tastiera per eseguire un programma Basic dall'interno di un altro

Listato 3

```
10  [[
20  :
30  DEMO ISTRUZIONI GRAFICHE
40  :
50  ]]
60  COLOR 1,0,0:T=0
70  REPEAT
80  FILL 0+T,0+T,40-2*T,25-2*T,160,T
90  T=T+1
100 UNTIL T=13
110 EXEC PAUSA
120 T=0
130 REPEAT
140 SUP 0,0,20,13,0
150 T=T+1
160 UNTIL T=14
170 T=0
180 REPEAT
190 SRIGHT 20,0,20,13,0
200 T=T+1
210 UNTIL T=21
220 T=0
230 REPEAT
240 SLEFT 0,13,20,12,0
250 T=T+1
260 UNTIL T=21
270 T=0
280 REPEAT
290 SDOWN 20,13,20,12,0
300 T=T+1
310 UNTIL T=13
320 EXEC PAUSA
330 COLOR 1,6,6
340 C=1
350 FORT=0 TO 1 :FORY=0 TO 3
360 FILL 4+ 8*Y,2+18*T,6,3,160,0:FILL 3+ 8*Y,1+18*T,6,3,160,C:C=C+1
370 IF C=6 THEN C=7
380 NEXT: NEXT
390 FILL 10,8,20,8,160,0
400 FILL 9,7,20,8,160,15
410 COLOR 15,6,6
420 LOCATE 13,9," RULTRA BASIC":LOCATE 13,11,"R" DEMO ""
430 EXEC PAUSA
440 END
450 PROC PAUSA
460 GETKEY AS
470 END PROC
```

*Listato 3.
Questo listato è da studiare contiene quasi tutte le istruzioni grafiche di questa puntata.*



programma. Inoltre Chain permette di implementare in maniera semplice e del tutto trasparente all'utente la tecnica degli overlay.

- **Gsave:** salva la pagina grafica (compreso anche il colore). La sintassi dell'istruzione è identica a quella dell'istruzione Dsave.

- **Gload:** carica una pagina grafica (compreso anche il colore). La sintassi dell'istruzione è identica a quella dell'istruzione Dload.

- **Bload:** questa è una delle istruzioni più utili del gruppo. Permette infatti di caricare un programma a partire da un determinato indirizzo e quindi è particolarmente utile per gestire in modo efficiente il caricamento di routine in linguaggio macchina, pagine grafiche, le definizioni degli sprite e dei caratteri.

La sintassi dell'istruzione è: `bload"nomefile"[,st]`.

Le parentesi quadre indicano che il parametro è opzionale: se questo non viene specificato l'istruzione si comporta come Dload, mentre, se c'è il programma viene caricato a partire dall'indirizzo indicato.

Per esempio con Bload monitor il programma monitor verrà caricato nel suo indirizzo d'origi-

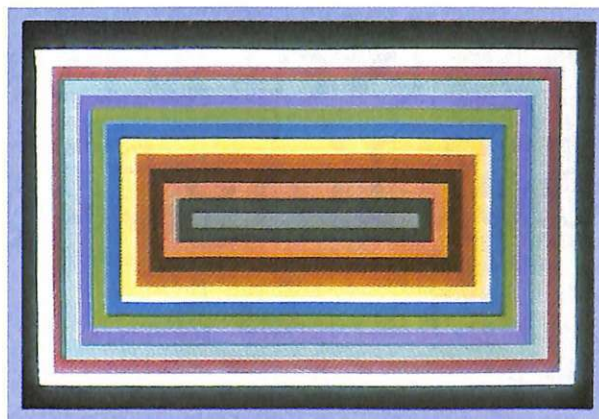


*Figura 1.
Videata
dal demo
del listato 3.
Pochissime
istruzioni per
realizzarla.*

ne, mentre con Bload monitor, 12000 il monitor sarà caricato a partire dall'indirizzo decimale 12000.

L'indirizzo può essere specificato sia in notazione decimale che sia in notazione esadecimale. In quest'ultimo caso è possibile vedere il numero che viene preceduto dal carattere \$.

- **Bsave:** permette di salvare su disco una porzione di memoria. La sintassi dell'istruzione è: `bsave"nomefile", st, en`.

I parametri st e en rappresentano rispettivamente l'indirizzo iniziale e quello finale + 1 del blocco di memoria che deve essere salvato.

Anche in questo caso gli indirizzi possono essere espressi sia in decimale sia in esadecimale.

- **Csave:** salva su disco la definizione dell'intero set di caratteri. La sintassi è identica a quella dell'istruzione Dsave.

- **Cload:** carica da disco un set di caratteri precedentemente salvato con Csave. La sintassi è identica a quella dell'istruzione precedente.

- **Dir:** visualizza la directory del disco nel drive. La visualizzazione dei file può essere bloccata temporaneamente premendo il tasto Shift.

Per far proseguire la visualizzazione ovviamente basterà rilasciare il tasto.

- **Disk:** permette di inviare un comando al drive. La sintassi dell'istruzione è: `disk c$` dove c\$ è la stringa di comando.

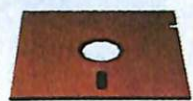
Per esempio per cancellare il file pippo dal disco nel drive basterà usare l'istruzione: `disk "s:pippo"`. Questa istruzione equivale all'istruzione del Basic standard: `open 1,8,15,"s:pippo":close 1`.

- **Error:** visualizza la stringa di errore del Dos (è utilissima per scoprire il mistero della luce lampeggiante del drive). L'istruzione non necessita di alcun parametro.

Daniele Maggio
(continua)



*Figura 2.
Schermata
tratto dal demo
del listato 3.
Questa videata
è stata
realizzata da
una sola linea
di programma*

**C64****SOFTWARE**

Nel vivo del linguaggio

Iniziamo finalmente ad addentrarci nel vivo del linguaggio. In questa puntata sono di scena infatti i tipi predefiniti, le variabili e gli operatori di Qpl.

Con questi elementi potete gustare le prime soddisfazioni di chi impara un nuovo linguaggio

Quarta puntata

Nelle puntate precedenti si è parlato soprattutto dell'ambiente di sviluppo e non del linguaggio vero e proprio. Come ricorderete è stato descritto il funzionamento dell'interprete di comandi e dell'editor. Purtroppo era necessario adottare un approccio di questo tipo perchè in caso contrario adesso non sapreste come scrivere un programma e quindi non sareste in grado di mettere in pra-

tica quanto verrà detto in questa e nelle prossime puntate. Ora che sapete come muovervi nell'ambiente integrato è possibile vedere più da vicino le caratteristiche del linguaggio Qpl.

Gli elementi del linguaggio

Nelle puntate precedenti è stato descritto il funzionamento dell'editor, lo strumento che vi permetterà di scrivere programmi nel linguaggio Qpl. È molto importante che lo sappiate usare con sufficiente disinvoltura perchè tra poco ne avrete bisogno. Prima di proseguire è necessario ricordare una cosa molto importante: Qpl è un linguaggio compilato e

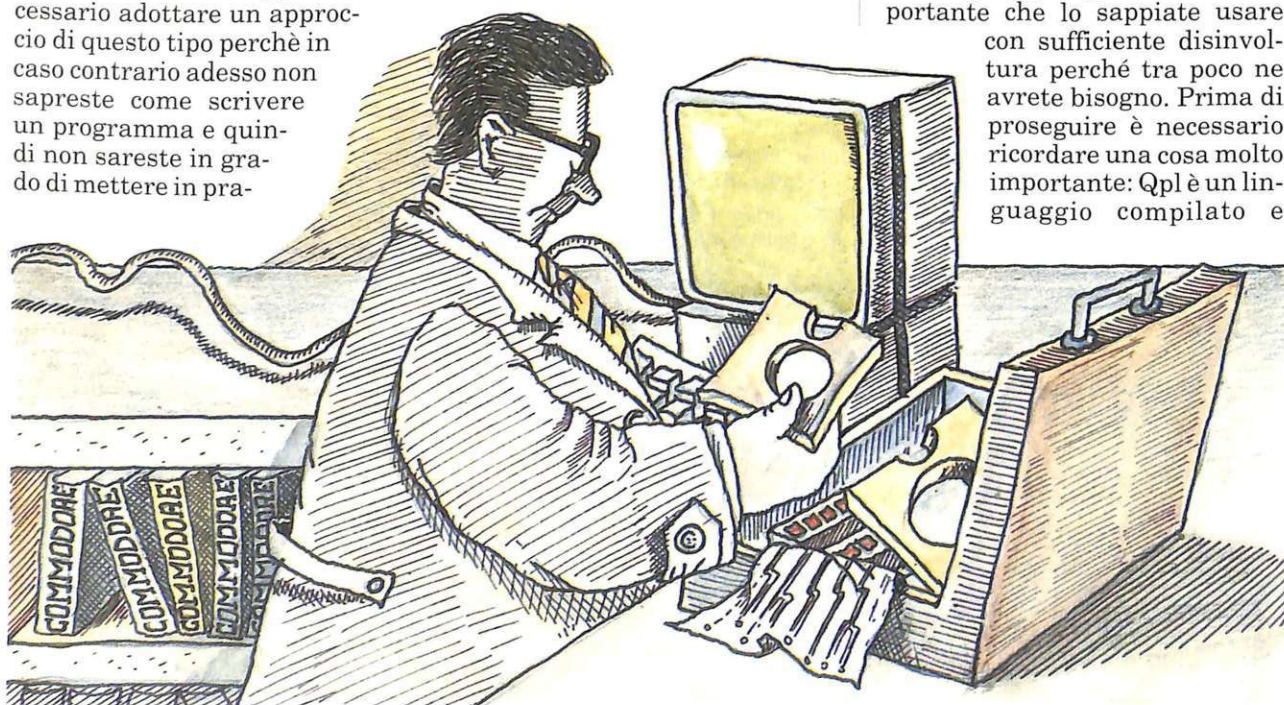




Tavola 1.

Procedura da seguire per compilare un programma. Da notare che ogni altro programma presente in memoria deve essere rimosso per fare spazio al compilatore:

- 1) salvare il programma su disco;
- 2) uscire dall'editor salvando il sorgente nel workspace;
- 3) rimuovere ogni programma dalla memoria;
- 4) digitare il comando Compile.

quindi ogni programma che scriverete per essere eseguito dovrà prima essere elaborato dal compilatore. Nella **tavola 1** viene sommariamente descritto il procedimento da seguire per compilare un programma Qpl. Del compilatore si parlerà più dettagliatamente in futuro. Potrete seguire lo schema per provare tutti gli esempi che verranno descritti nel corso delle prossime puntate e anche per fare qualche esperimento di programmazione. Nella **tavola 2** viene riportato l'elenco completo delle parole chiave del linguaggio. Come potete notare il Qpl risulta decisamente più ricco del Basic standard (anche se in realtà non sarebbe molto corretto confrontare due linguaggi solo dai rispettivi vocabolari). Nell'elenco tutte le parole chiave iniziano con un carattere maiuscolo seguito da caratteri minuscoli. Nei programmi, tuttavia, le parole chiave potranno essere scritte utilizzando qualsiasi combinazione di caratteri maiuscoli e minuscoli. Per esempio in un programma potrete scrivere indifferentemente AND, and o and. Tene presente che in Qpl, a differenza del Basic, le parole chiave devono essere separate le une dalle altre con degli spazi. Pur essendo fondamentali, le parole chiave non sono gli unici elementi che costituiscono un programma. Un'altra classe di oggetti che entra in gioco nella stesura di ogni programma è costituita dagli identificatori o nomi.

Gli identificatori o nomi

I nomi in Qpl servono per identificare costanti, variabili, funzioni, procedure e gli stessi programmi. Tutti gli identificatori devono essere formati utilizzando le seguenti regole:

- 1) un nome può essere lungo da un minimo di un carattere a un massimo di 31;
- 2) il primo carattere di un nome deve essere una lettera;
- 3) tutti gli altri caratteri devono essere lettere, numeri o il carattere freccia a sinistra;
- 4) si possono usare sia caratteri maiuscoli sia minuscoli. Entrambi sono considerati equivalenti (quindi l'identificatore xyz sarà considerato equivalente agli identificatori XYZ e XyZ);
- 5) un nome non può essere uguale a una parola riservata.

A differenza del Basic in Qpl tutti i caratteri presenti nell'identificatore sono significativi. Per esempio, appartamento1 e appartamento 2 sono considerati due identificatori distinti anche se hanno i primi 12 caratteri uguali. Gli identificatori occupano tutti la stessa quantità di me-

moria e quindi usando identificatori brevi (perciò poco significativi) non si risparmia memoria rispetto a quando si usano identificatori più lunghi. Per essere più precisi, la differenza è solo a livello di sorgente e non di codice oggetto; infatti è solo il testo del programma che risulta più lungo se si usano identificatori più lunghi e non il codice oggetto.

Tavola 2.

Elenco completo delle parole chiave del Qpl:

```
And Con Ext Int Own To
Arg Chose False List Proc True
Asm Data For Next Program Until
At End Funk Not Real While
Byte Else If Nothing Refuge Word
Begin Escape Import Or Repeat XOR
Break Export Include Overlay Return
```

Nella **tavola 3** vengono mostrati alcuni esempi di identificatori corretti e non. Un altro aspetto importante del linguaggio è l'insieme dei tipi di dati predefiniti.

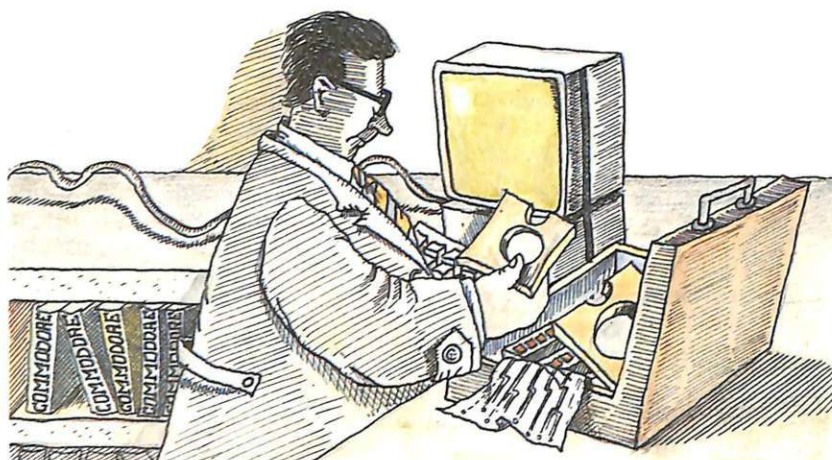
Tipi predefiniti

Il tipo è un attributo che ciascun dato, variabile o costante, possiede e identifica l'insieme dei valori che il dato può assumere, e l'insieme delle operazioni che possono essere eseguite su di esso. Nella **tavola 4** viene riportato l'elenco dei tipi disponibili in Qpl con il relativo significato.

Le variabili di tipo byte, word e int non possono contenere parti frazionarie. Quindi assegnare il valore 11 o 15 a una variabile int è corretto, ma non lo è assegnare il valore 12.8. Le variabili real occupano sei byte e hanno 11 cifre significative (in Basic le variabili di questo tipo occupano cinque byte e hanno solo nove cifre significative).

**C64E**

SOFTWARE



Le variabili di questo tipo sono manipolate molto più lentamente di quelle degli altri tipi disponibili, e quindi vanno usate solo quando è strettamente necessario. Tutti i numeri scritti senza il punto decimale vengono interpretati come non real (cioè possono essere di un tipo qualsiasi tranne real) mentre quelli in cui compare il punto decimale sono considerati real e, quindi trattati di conseguenza.

Inoltre i numeri possono essere specificati sia in notazione decimale sia in notazione esadecimale (in questo caso il numero va preceduto dal carattere \$).

Nella **tavola 5** sono riportati alcuni esempi di numeri di vario tipo corretti e non. Da notare che i numeri reali possono anche essere espressi in notazione scientifica.

Stringhe e caratteri

In Qpl come in Basic una costante stringa è una sequenza di caratteri racchiusa fra doppi apici. Esempi di stringhe sono: "a", "asdfg", "123", il segno di addizione, sottrazione, moltiplicazione, divisione ("+", "-", "x") e la stringa vuota (""). Una costante stringa occupa esattamente tanti byte quanti sono i caratteri che

la compongono + 1. Il byte aggiuntivo viene utilizzato dal terminatore, uno zero, che viene automaticamente aggiunto a ogni stringa. Per questo motivo le stringhe vuote ("") e "1" occuperanno rispettivamente uno e due byte. Potete inserire caratteri di controllo nelle stringhe utilizzando il carattere £ seguito da due cifre esadecimali che rappresentano il codice del carattere di controllo. Ecco un esempio: put "£12 pippo" (put è una procedura predefinita del Qpl che equivale al print del Basic) visualizza in reverse la stringa pippo (£12 è il codice del carattere rvs on). Per inserire il carattere di doppio apice (") in una stringa dovete scriverlo due volte di seguito. Ecco un esempio: put "egli disse: ""torno subito""".

Per specificare un costante ca-

ratte re dovete usare gli apici singoli. Ecco qualche esempio: 'a', '0' e segno di addizione ('+').

Per specificare il carattere di apice (') dovete scriverlo due volte di seguito: ''.

Una nota molto importante: la costante carattere 'a' è diversa dalla costante stringa "a"; infatti la costante 'a' occupa un solo byte e a essa il compilatore attribuisce il valore 65 (il codice Ascii del carattere a), mentre "a" occupa due byte di memoria e a essa il compilatore associa un puntatore alla locazione di memoria in cui viene posto il carattere a (questo concetto verrà ulteriormente approfondito in futuro).

Le variabili e le costanti

In Qpl tutte le variabili devono essere dichiarate prima di essere usate. Una dichiarazione serve a indicare al compilatore il nome della variabile e il suo tipo (e quindi implicitamente anche quanta memoria occupa).

Ecco un esempio di dichiarazione: int punteggio.

Questa dichiarazione serve per creare una variabile di tipo int (quindi una variabile destinata a contenere un valore intero compreso fra 32767 e -32767) e assegnarle il nome punteggio. Rispetto al Basic c'è quindi l'obbligo di dichiarare preventivamente tutte le variabili che si vogliono usare. Per alcuni questo costituirà una notevole seccatura all'inizio

Tavola 3.

Esempi di identificatori corretti e non.

Identificatori corretti: A, Zero, Prova, For4, d2000, c4

Identificatori non corretti: b-4 (il secondo carattere non è né alfanumerico né con la freccia a sinistra)

3d (il primo carattere deve essere una lettera) List (è una parola chiave)



Tavola 4.

Tipi predefiniti in Qpl.

tipo	significato
byte	intero senza segno compreso fra 0 e 255, oppure un carattere
Ascii,	oppure il valore booleano vero o falso
word	intero senza segno compreso fra 0 e 65535
int	intero con segno compreso fra -32767 e 32767
real	numero reale compreso, approssimativamente, fra $1.e-37$ e $1.e+37$.

ro di elementi presenti in quella dimensione. Ecco come si dichiarano degli array multidimensionali:

byte schermo [40,25]
byte a1 [10,8,3]

L'array schermo è bidimensionale, mentre a1 è tridimensionale. La memoria occupata da un array è pari al prodotto degli elementi di ogni dimensione per

ma, con il tempo vedrete che vi aiuterà a scrivere programmi molto ordinati e quindi leggibili. Un'altra importante differenza rispetto al Basic è che le variabili in Qpl non vengono mai iniziate automaticamente.

Ciò significa che il contenuto di una variabile mai usata deve essere considerato assolutamente casuale.

Una costante è un nome associato a un valore numerico che non muta durante l'esecuzione del programma. Per definire una costante si deve usare la parola chiave con, come in questo esempio: con lf=10; questa linea permette di associare alla costante lf il valore 10.

Dopo aver incontrato questa linea il compilatore Qpl sostituirà ogni occorrenza del nome lf con il valore 10.

Il tipo della costante viene automaticamente fissato dal compilatore se non viene indicato altrimenti. Volendo, infatti, potete specificare anche il tipo della costante.

Ecco un esempio: con word start=\$40; la costante start è di tipo word, ma sarebbe di tipo byte (\$40 è un intero compreso fra zero e 255) se non comparisse word nella dichiarazione. Esiste una sola limitazione nella dichiarazione del tipo di una costante: non è possibile dichiarare delle costanti reali.

Per definire delle costanti di

Tavola 5.

Alcuni esempi di numeri espressi in forma corretta e non. I numeri reali possono essere espressi anche in notazione scientifica.

Numeri di tipo byte, int o word corretti: 0, 1, 137, 22340, 65535, -78

Numeri non corretti: 1,333 (si deve usare il punto e non la virgola) 120.4 (è legale solo se viene assegnato a una variabile real ma non se viene assegnato a una variabile di altro tipo) 65579 (va oltre il limite superiore)

questo tipo dovete servirvi dello statement data (sarà descritto in futuro).

Array

La definizione degli array avviene in modo abbastanza simile al Basic. Un array può essere di uno qualsiasi dei quattro tipi predefiniti. Il numero massimo delle dimensioni di un array è otto. Ecco qualche esempio di dichiarazione di array:

```
con bufsize=100
byte buffer1 [bufsize]
byte buffer2 [bufsize]
```

Buffer1 e buffer2 sono due array monodimensionali di 100 elementi di tipo byte. Buffer1[0] contiene il primo elemento di buffer1 e buffer[99] l'ultimo. Per ogni dimensione l'indice di un array parte sempre da zero e arriva sino a dim-1 dove dim è il nume-

l'occupazione dell'elemento base dell'array. Per esempio, l'array schermo occupa 1000 byte mentre a1 occupa 10 x 8 x 3 byte.

Le costanti data

Nella dichiarazione delle costanti non possono comparire degli array e non si possono inserire dei valori reali.

Per superare questo limite il Qpl mette a disposizione le costanti data. Ecco come si può usare questo strumento:

```
data real pi = 3.1415
data byte tabella [] = 23, 52, 1, 9-14
```

Nel primo caso alla costante pi viene associato un valore reale mentre nel secondo caso viene inizializzato un intero array. Questo meccanismo è particolarmente utile per definire vettori di stringhe. Ecco un esempio:

**C64****SOFTWARE**

data word messaggio err[]=
 "file inesistente",
 "disco pieno",
 "file già aperto"

L'ultimo esempio è particolarmente importante. Si può notare che l'array è stato dichiarato di tipo word e non byte. Questo perché ogni elemento del vettore è una stringa e il Qpl identifica ogni stringa con un puntatore al primo carattere e un puntatore, occupando due byte, rende necessario l'uso di word (in Qpl la gestione delle stringhe avviene in modo sostanzialmente simile al C. Ritorneremo più avanti sull'argomento più in dettaglio).

Operatori

Un operatore è un simbolo speciale che indica un'azione da compiere. Tutti gli operatori disponibili in Qpl sono elencati nella **tavola 6**.

Alcuni di questi sono sicuramente famigliari a chi conosce il Basic ma altri sono completamente nuovi.

La priorità degli operatori Qpl è riassunta nella **tavola 7**. Ve-

Tavola 6.

Operatori disponibili in Qpl. Non c'è confronto rispetto al Basic.

operatore:	significato:
+	addizione
-	sottrazione
*	moltiplicazione
/	divisione
%	resto
<<	shift a sinistra
>>	shift a destra
<	minore
<=	minore o uguale
<>	diverso
=	uguale
>=	maggiore o uguale
>	maggiore
And	and
Or	or
Xor	or esclusivo
Not	not
#	indirizzo di variabile
:<	estrae byte basso
:>	estrae byte alto
:+	converte in word
:-	converte in int
::	converte in real
@<	accede a una variabile di tipo byte tramite puntatore
@-	accede a una variabile di tipo int tramite puntatore
@+	accede a una variabile di tipo word tramite puntatore
@.	accede a una variabile di tipo real tramite puntatore

Tavola 7.**Tabella della priorità degli operatori**

:<, :>, :+, :-, ::, @<, @+, @-, @., # priorità maggiore
 not
 *, /, %, <<, >>
 - (unario)
 +, -
 <, <=, <>, =, >=, >
 and, or, xor

priorità inferiore

diamo nei dettagli come si comportano gli operatori:

• **addizione, sottrazione, moltiplicazione, divisione, (+, -, *, /):** sono i noti operatori aritmetici fondamentali di addi-

zione, sottrazione, moltiplicazione e divisione (produce un risultato intero se gli operandi sono interi).

• **segno di percentuale (%):** è l'operatore di resto. Per esem-

pio, 200 % 30 produce 20, appunto il resto della divisione di 200 per 30.

• **<< e >>:** sono gli operatori di scorrimento o Shift: << è l'operatore di Shift a sinistra mentre >> e quello di Shift a destra. Due esempi ne chiariranno il funzionamento:

7 << 1 da come risultato 14
 \$a0 >> 4 da come risultato \$0a

• **<, >, <=, >=, <>, =:** sono gli operatori relazionali. Un'espressione contenente solo operatori relazionali viene detta espressione booleana e fornisce un valore vero o falso. In Qpl falso equivale a zero mentre, vero equivale a non zero. Le due costanti predefi-



C64

nite true e false possono essere usate al posto dei valori numerici e valgono rispettivamente uno e zero. Ecco qualche esempio:

4 < 9 risultato true
6 <= 6 risultato true
'a' <> 'a' risultato false
'a' = 65 risultato true
10 >= 'a' risultato false
3 > 8 risultato false

• **And, or, xor, not:** sono gli operatori logici.

Anche in questo caso alcuni esempi basteranno per chiarire il funzionamento:

3 > 1 and 4 < 10 risultato true
2 <= 1 or 8 > 9 risultato false

\$00 xor \$ff risultato \$ff
not true risultato false

• **#:** permette di conoscere l'indirizzo di una variabile, cioè applicato a una variabile qualsiasi ne fornisce l'indirizzo. Ecco un esempio: #x risultato indirizzo della prima locazione dedicata alla variabile x.

• **segno di minore (<):** questo segno estrae il byte basso da un valore word o int. Ecco un esempio: \$1234:< risultato \$34.

• **segno di maggiore (>):** permette di estrarre il byte alto da un valore word o int. Ecco un esempio: \$1234:> risultato \$12.

• **segno di addizione (+):** converte in word. Ecco un esempio: \$5a:+ risultato \$005a.

• **segno di sottrazione (-):** converte in int. Ecco un esempio: \$ff:- risultato +255.

• **segno di punto(.):** converte in real. Ecco un esempio: 45:. risultato 45.0.

Gli ultimi quattro parametri sono gli operatori di direzione e servono per reperire un oggetto referenziato da un puntatore ma di questo parleremo nella prossima puntata.

Daniele Maggio
(continua)

LOGITEK PRESENTA:

Sistemi completi per
DeskTop Video e
DeskTop Publishing.

**Assortimento completo
di periferiche grafiche
per Amiga:**

- scanner manuali e A4
- genlock e digitalizzatori professionali e amatoriali
- splitter
- tavole grafiche
- hard disk ed espansioni
- allestimento stazioni complete per la grafica, la titolazione e il publishing.

OFFERTA STREPITOSA

Stampante professionale
Commodore, 136 colonne, per PC
e compatibili, Amiga, Commodore
64, ottima per stampare testi

Lire **229.000*** iva compresa

* Offerta valida fino
esaurimento scorte.



Davanti agli schermi dei computer a volte si rimane a bocca aperta e senza fiato. Anche il Commodore 64 ha questo potere nei confronti dell'osservatore e i tips di questo mese ne sono la dimostrazione

Un kit per... mozzare il fiato!

Con poche linee Basic, come è stato spesso dimostrato in questa rubrica, si possono realizzare effetti grafici molto interessanti che possono servire anche come spunto per programmi molto più complessi.

Kaleidoscope

Un classico esempio è Kaleidoscope. Con una decina di linee Basic (potrebbero essere molto

meno se si fosse evitato di presentare il listato in maniera leggibile) è stato possibile realizzare, anche se in forma abbastanza approssimativa, il noto effetto kaleidoscopio.

Questo programma si presta a diversi miglioramenti e offre diversi spunti a chi ha qualche interesse nella grafica.

Innanzitutto potreste provare a compilare il programma per vedere l'effetto che si ottiene con

una esecuzione molto più rapida. Chi conosce il linguaggio macchina potrebbe anche tentare di tradurlo in assembler (è un ottimo esercizio perché potrete scoprire il modo di generare numeri casuali in lm). Per lanciare il programma non dovete far altro che copiare il **listato 1** e dare il Run. Durante l'esecuzione potrete cancellare lo schermo premendo la barra spaziatrice.

Kaleidoscope può essere facilmente inserito nei vostri programmi con pochissimo sforzo: l'unica modifica che è possibile apportare al programma riguarda il numero di linea che compare dopo il Goto alla linea 200.

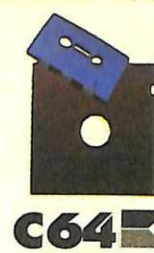
Listato 1. Kaleidoscopio

```
1 rem -----
2 rem -
3 rem - kaleidoscope -
4 rem -
5 rem -----
6 :
10 print"<CLEAR>";chr$(142)::z=160
20 poke 53280,0:poke53281,0
25 :
30 c=int(rnd(0)*15)
40 x=int(rnd(0)*40)
50 y=int(rnd(0)*25)
60 :
70 e1=1024:e2=1063:e3=1984:e4=2023
80 f1=55296:f2=55335:f3=56256:f4=56295
90 :
100 poke e1+x+40*y,z
110 poke f1+x+40*y,c
120 poke e2-x+40*y,z
130 poke f2-x+40*y,c
140 poke e3+x-40*y,z
150 poke f3+x-40*y,c
160 poke e4-x-40*y,z
170 poke f4-x-40*y,c
180 :
190 if peek(203)=60 then print"<CLEAR>";
200 goto 30
```

Effetti speciali

Con questa seconda serie di effetti speciali potrete arricchire degnamente tutti i vostri programmi, di qualsiasi genere siano, e stupire chi li vede in esecuzione. Come al solito le routine che li realizzano sono scritte interamente in linguaggio macchina e sono strutturate in modo tale da non creare grossi problemi di compatibilità o di adattamento ai vostri programmi.

Tutte le routine vengono infatti allocate in zone di memoria al



riparo dal Basic e di dimensioni estremamente limitate.

I listati delle routine sono contenuti nei **listati 2, 3, 4 e 5** sono già pronti per essere incorporati nei vostri programmi.

L'unica modifica che dovete apportare è l'eliminazione della Sys di attivazione che dovrà ovviamente comparire in una linea del vostro programma.

Ecco una brevissima descrizione degli effetti grafici che avete a vostra disposizione. Special effect 1 (s.e.1) realiz-

za una specie di effetto nebbia con righe sottilissime di vario colore, che si muovono molto velocemente verso l'alto.

Special effect 2 (s.e. 2) è un tipico effetto da video game perchè è davvero molto spettacolare.

Special effect 3 (s.e 3) ricorda vagamente l'effetto grafico in Alien che segnala l'aggressione dell'alieno. Special effect 4 (s.e 4) è tra tutti l'effetto più serio, e

Listati da 2 a 5. Effetti speciali

```
1 rem -----
2 rem - -
3 rem - s.e. 1 -
4 rem - -
5 rem -----
6 :
10 fort=49152 to 49205:read d:poke t,d:next
20 sys 49152
30 :
40 data 120,173,32,208,72,173,17,208,41
50 data 239,141,17,208,162,0,142,32,208
60 data 173,1,220,201,255,208,14,138,105
70 data 4,170,181,2,234,224,255,208,235
80 data 76,13,192,173,17,208,9,16,141
90 data 17,208,104,141,32,208,88,96,9
```

```
1 rem -----
2 rem - -
3 rem - s.e. 2 -
4 rem - -
5 rem -----
6 :
10 fort=49152 to 49205:read d:poke t,d:next
20 sys 49152
30 :
40 data 120,173,32,208,72,173,17,208,41
50 data 239,141,17,208,162,0,142,32,208
60 data 173,1,220,201,255,208,14,232,232
70 data 232,232,232,232,232,224,255,208,235
80 data 76,13,192,173,17,208,9,16,141
90 data 17,208,104,141,32,208,88,96,9
```

```
1 rem -----
2 rem - -
3 rem - s.e. 3 -
4 rem - -
5 rem -----
6 :
10 fort=49152 to 49201:read d:poke t,d:next
20 sys 49152
30 :
40 data 120,173,32,208,72,173,17,208,41
50 data 239,141,17,208,162,0,142,32,208
60 data 173,1,220,201,255,208,10,232,234
70 data 234,224,16,208,239,76,13,192,173
80 data 17,208,9,16,141,17,208,104,141
90 data 32,208,88,96,19,2
```

```
1 rem -----
2 rem - -
3 rem - s.e. 4 -
4 rem - -
5 rem -----
6 :
10 fort=49173 to 49198 step
5:poket, rnd(1)*256:next
20 fort=49152 to 49226:readd:poke t,d:next
30 sys 49152
40 :
50 data 120,173,32,208,72,173,17,208,41
60 data 239,141,17,208,173,1,220,201,255
70 data 208,40,169,0,141,32,208,169,2,141
80 data 32,208,169,3,141,32,208,169,5,141
90 data 32,208,169,6,141,32,208,169,7,141
100 data 32,208,234,230,2,230,2,165,2,76
110 data 13,192,173,17,208,9,16,141,17
120 data 208,104,141,32,208,88,96,96
```

Listato 6. Smooth scrolling

```
1 rem -----
2 rem - -
3 rem - smooth scrolling -
4 rem - -
5 rem -----
6 :
40 h$="0123456789abcdef"
50 hd$=""
60 v=52223
70 read hd$:v=v+1:printv,hd$
80 if v=52435 then goto 500
90 if len(hd$)=0 then print"errore":end
100 if len(hd$)<>2 then print"errore":end
110 h=0
120 fori=1to16
130 if left$(hd$,1)=mid$(h$,i,1) thenh=i:i=16
140 next i
150 if h=0 then poke v,0:goto 70
160 h=h-1
170 l=0
180 fori=1 to 16
190 if right$(hd$,1)=mid$(h$,i,1) then l=i:i=16
200 next i
210 if l=0 then poke v,0:goto 70
220 l=l-1
230 d=h*16+1
240 pokev,d
250 goto 70
255 :
256 :
260 data 78,a9,3e,8d,14,03,a9,cc,8d
270 data 15,03,ad,11,d0,29,7f,8d,11
280 data d0,a9,90,8d,12,d0,a9,81,8d
290 data 19,d0,8d,1a,d0,ad,0e,dc,29
300 data fe,8d,0e,dc,ad,16,d0,29,f7
310 data 09,07,8d,16,d0,a9,e0,85,fb
320 data a9,cc,85,fc,c6,fb,58,60,ad
330 data 19,d0,29,01,f0,23,c6,fd,10
340 data 04,a9,07,85,fd,c6,fd,ad,16
350 data d0,29,f0,05,fd,8d,16,d0,a9
360 data 79,8d,14,03,a9,cc,8d,15,03
370 data a9,a4,8d,12,d0,a9,81,8d,19
380 data d0,8d,1a,d0,ad,0d,dc,68,a8
390 data 68,aa,68,40,ad,19,d0,29,01
400 data f0,48,ad,16,d0,09,07,8d,16
410 data d0,a5,fd,d0,2d,a8,c8,b9,08
420 data 06,88,99,08,06,4c,9a,cc,ea
430 data ea,c8,c0,28,d0,ee,a0,00,e6
440 data fb,d0,02,e6,fc,b1,fb,d0,0b
450 data a9,e0,85,fb,a9,cc,85,fc,4c
460 data a7,cc,8d,2f,06,a9,3e,8d,14
470 data 03,a9,cc,8d,15,03,a9,90,8d
480 data 12,d0,a9,81,8d,19,d0,8d,1a
490 data d0,4c,31,ea,00
500 :
510 a=52448:poke204,0:print"stringa:"
520 get a$:ifa$="" then 520
530 if a$=chr$(13) then pokea,0:end
540 if asc(a$)<65 then
printa$::pokea,asc(a$):a=a+1:goto520
550 printa$::pokea,asc(a$)-64:a=a+1:goto520
```




COMMODISK

l'unica rivista con dischetto per Commodore 64 e C128

Listato 7. Quote switcher

```

1 rem -----
2 rem -
3 rem - quote switcher -
4 rem -
5 rem -----
6 :
10 t=0:sa=49152
20 forc=sa to sa+37:readd:t=t+d:pokec,d:next
30 if t<>4252 then print"<CLEAR>errore nei data":end
40 sys 49152
50 print"quote switcher installato":end
60 :
70 data 120,169,13,141,20,3
80 data 169,192,141,21,3,88
90 data 96,165,197,201,5,208
100 data 7,169,0,133,212,76
110 data 34,192,201,4,208,4
120 data 169,1,133,212,76,49
130 data 234,186

```

per questo motivo si presta benissimo per coprire le pause di un programma.

Smooth scrolling

Restando sempre in tema di grafica, uno degli strumenti più suggestivi e fantasiosi fra quelli messi a disposizione dal C64 è lo scrolling. Questo spettacolare effetto grafico viene usato raramente da chi programma in Basic appunto perché richiede la stesura di una routine di supporto in linguaggio macchina per poter essere utilizzato.

Smooth scrolling dovrebbe risolvere definitivamente questo problema perché vi permette di visualizzare stringhe con effetto scrolling senza costringervi a realizzare complicate routine in lm. Inoltre Smooth scrolling può essere inserito molto facilmente nei vostri programmi con pochissime modifiche. Per poter utilizzare la routine dovete copiare il **listato 6**, dare il Run e digitare la stringa da visualizzare con effetto scrolling.

Quando volete iniziare lo show non dovete far altro che digitare Sys 52224. Due note importantissime: per ottenere caratteri in Reverse basta shiftarli, per inserire una nuova stringa dovete far eseguire la linea 500.

Quote switcher

Del Quote mode ormai dovrete sapere tutto. Si tratta del famigerato modo operativo dell'editor che determina la visualizzazione di strani caratteri quando, dopo aver digitato una volta il carattere doppie virgolette ("), tentate di utilizzare i tasti di movimento del cursore.

Una volta che è stato attivato, il Quote mode può costituire un problema perché limita notevolmente la vostra libertà di movimento. La soluzione migliore a questo tipo di problema è senza dubbio Quote switcher.

Questa micro utility (occupa una manciata di byte nel vero senso della parola) vi permetterà di attivare e disattivare a vostro piacimento il Quote mode rendendo quindi l'editor del Commodore 64 estremamente più comodo e professionale.

Per utilizzare la routine dovete copiare il **listato 7** e premere di nuovo il tasto Run.

Dopo alcune frazioni di secondo sul video verrà visualizzata la scritta: Quote switcher attivato. A questo punto disponete di due comandi:

- **F1**: attiva il Quote mode.
- **F3**: disattiva il Quote mode.

Daniele Maggio



Come tu vuoi!

Ecco una panoramica sugli accessori che permettono all'Amiga 2000 di concorrere con tutti i più diffusi e potenti personal computer sul mercato. Parliamo di schede interne, per emulare Ibm e per dare sprint all'elaborazione di Amiga, e di una stampante di grande successo

Anche se Amiga è un personal computer tra i più sofisticati esistenti sul mercato, non c'è dubbio che la scelta di un sistema operativo autonomo comporta non pochi problemi per gli utenti. Questi rimangono infatti confinati

nell'ambito di tale sistema, senza possibilità di accedere alle centinaia di programmi esistenti, per esempio, in versione Ms-Dos.

Per ovviare a questo inconveniente la Commodore ha realizzato in tempi successivi sistemi o

schede aggiuntive, capaci di rendere Amiga un computer multiplo. Infatti, a differenza di altri personal in grado di emulare l'Ms-Dos, Amiga lo vede come una task, cioè come uno dei possibili ambienti operativi con cui la-





AMIGA È



vorare. In pratica, è quindi possibile gestire contemporaneamente i due sistemi. Ma procediamo con ordine.

Sistemi come il Sidecar, atti a trasformare Amiga in un computer Ms-Dos compatibile, non sono nuovi per la Commodore. Per poter realizzare questa trasformazione si è però dovuto operare in un modo piuttosto originale: inserendo, esternamente o internamente ad Amiga, un vero e proprio computer XT o AT su singola scheda. Questo tipo di scelta, sebbene garantisca un'elevata compatibilità con il sistema emulato, non può ovviamente essere a buon mercato: i costi dei vari dispositivi aggiuntivi sono infatti paragonabili a quelli di un normale computer XT o AT "made in Taiwan".

Lavorare in multitasking

Com'è noto, più di un'applicazione può girare contemporaneamente su Amiga. Nel nostro caso l'ambiente Ms-Dos viene considerato dal sistema come una task, e pertanto applicazioni Ms-Dos possono girare contemporaneamente ad applicazioni Ami-

ga. Questo rende estremamente interessante il computer sotto molti punti di vista, ma genera anche un non semplice problema quando è richiesta una buona velocità di elaborazione. Infatti, se già un XT non è certo velocissimo, Amiga rallenta ulteriormente i tempi di esecuzione.

Attualmente la Commodore ha messo in circolazione una scheda AT denominata Bridgeboard, che potremmo interpretare come ponte fra hardware diversi, o meglio fra sistemi operativi differenti, proprio per quelle caratteristiche di multiambiente a cui abbiamo accennato. La velocità di elaborazione di un AT è sensibilmente maggiore di quella di un XT, per cui il rallentamento provocato da Amiga si fa sentire meno.

Installazione

La confezione creata dalla Bridgeboard comprende, oltre alla scheda vera e propria, un drive da 5"1/4 da 1,2 Mb, una batteria ricaricabile, un floppy per l'inizializzazione della scheda dall'ambiente Amiga, e floppy

e manuali del sistema operativo Ms-Dos. L'installazione della scheda è semplice e prevede solo il suo inserimento in uno dei due possibili slot utilizzabili a questo scopo. Rimanendo liberi cinque slot, questi possono essere ripartiti come due per Amiga e tre per Ms-Dos o viceversa.

Successivamente si dovrà provvedere a installare il drive da 5"1/4, e effettuare il collegamento fra quest'ultimo e il controller montato sulla scheda tramite l'apposito cavo in dotazione. Da ultimo va installata la batteria ricaricabile, che serve ad alimentare l'orologio interno anche a computer spento. È anche possibile installare altre schede per l'ambiente Ms-Dos, quali per esempio, schede per grafica avanzata, fax, e modem.

Caratteristiche del sistema

La scheda AT per Amiga ha delle caratteristiche estremamente interessanti, a cominciare dalla possibilità di utilizzare nello stesso momento software di Amiga e software Ms-Dos (pur con le limitazioni sopra indicate).

Per il software Ms-Dos è possibile avvalersi per la rappresentazione a video del monitor di Amiga, il quale permette di visualizzare quattro colori nelle applicazioni che richiedono l'uso di schermo monocromatico, e 16 colori in quelle che utilizzano invece lo schermo a colori.

Particolarmente interessante è la possibilità di visualizzare contemporaneamente entrambe le emulazioni video, sia in finestre separate sia su diversi piani di lavoro. Applicazioni che permettono quindi l'uso di due monitor possono essere gestite sull'unico monitor di Amiga.

Non solo le applicazioni Ms-Dos possono essere gestite in modo insolito, ma anche applicazioni



Amiga trovano nella presenza della scheda AT numerosi vantaggi. La scheda può essere sfruttata come coprocessore per effettuare più velocemente, e in background operazioni di ricerca, ordinamento e altro ancora.

La possibilità di trasportare file tra applicazioni Ms-Dos e applicazioni Amiga permette poi di utilizzare la scheda AT e le relative interfacce o periferiche, per effettuare acquisizioni o trasferimento di dati elaborati con Amiga. Lo stesso discorso può in teoria essere applicato anche alle immagini elaborate su Amiga, e rappresentate su monitor ad alta definizione disponibili per ambiente Ms-Dos.

Software di sistema

Il manuale si dilunga sulla procedura d'installazione del software di sistema, operazione che nella pratica si rivela molto più semplice e chiara di quanto le lunghe spiegazioni non possano far pensare.

Il software d'installazione è contenuto in un dischetto di Amiga, in quanto la macchina deve

comunque partire dall'ambiente che le è più naturale: in pratica l'installazione del software consiste nella creazione di un dischetto contenente i file di un Workbench PC (i file possono essere installati su hard disk).

Non appena inserito il dischetto compaiono sul video le diverse icone che servono alla configurazione del sistema: la gestione dell'emulazione PC da Amiga è infatti piuttosto flessibile.

L'icona del PC compare fra quelle tipiche dell'ambiente Amiga e la sua selezione porta alla visualizzazione di altre nove icone: selezione del PC monocromatico, selezione del PC a colori, selezione della stampante, inizializzazione del mouse, inizializzazione dell'hard disk, inizializzazione disco parametri.

Particolarmente interessante è anche il successivo Setup, in cui è possibile variare la configurazione del sistema modificando date, tipi di drive e tipo di video, semplicemente ridigitando la propria scelta sulla tabella che appare a video.

Ma ciò che rende unico il modo di operare di questa scheda ri-

spetto ai tradizionali sistemi di emulazione è la possibilità di modificare numerosi parametri anche dall'ambiente Amiga utilizzando la tradizionale interfaccia utente a cui Commodore ci ha abituati (non solo icone, ma anche menù pop-down). Le possibilità anche in questo caso sono molteplici, ma, al di là della tradizionale scelta dei colori, delle dimensioni delle pagine, della velocità di reazione della tastiera, particolarmente interessante è la possibilità di aprire contemporaneamente più di una finestra su di un'applicazione, bloccando, se lo si desidera, l'immagine presente in una finestra e aprendone un'altra per continuare l'esecuzione del programma (non è possibile, ovviamente, avere più applicazioni Ms-Dos aperte nello stesso momento, in quanto questo sistema operativo non è multiutente).

Questa possibilità di apertura contemporanea di più finestre su di uno stesso programma è tipica di ambienti come Windows di Microsoft, che però è in grado di gestire solo un numero limitato di programmi. È quindi una novità

Tipo: scheda di emulazione Ibm PC AT utilizzante processore 80286 e opzionalmente coprocessore 80287

Velocità di clock: 8 MHz

RAM: 1 Mb

ROM: 16 Kb di BIOS

Interfacce:

1 per drive da 5 1/4 1,2 Mb

1 per drive da 3,5 720 Kb

1 per drive da 5 1/4 da 360 Kb

1 parallela emulata sulla porta parallela di Amiga

Slot d'espansione: 1 compatibile Ibm PC/AT; 2 compatibili Ibm PC/XT.

Tastiera: la tastiera Ibm viene emulata sulla tastiera di Amiga, compresa la disposizione dei tasti

Monitor supportati: MDA monocromatico, CGA a colori possono essere visualizzati contemporaneamente su finestre diverse.

Schede d'espansione supportate dall'ambiente Ms-Dos: AST Advantage Premium Hayes Smartmodem 1200 (modem interno), Mountain 20 Mb, Drive Card, Plus Hardcard, Taurus Lan Card, ATI EGA Wonder Orchid, VGA Card Paradise, VGA Plus Tecmar, VGA Card.

Altre caratteristiche: possibilità di gestire più drive internamente possibilità di ripartire, l'hard disk parte per l'ambiente Amiga e anche per l'ambiente Ms-Dos, supporto del mouse possibilità di gestire fino a cinque dischi virtuali



anche per gli utenti Ms-Dos l'opportunità di gestire, in teoria, una qualsiasi applicazione su più finestre contemporaneamente.

Ancor più sorprendenti sono le possibilità di taglia e incolla, con le quali è possibile togliere parti di testo presenti su una finestra e sistamarle su di un'altra.

Di estremo interesse è anche la possibilità di trascodificare file Amiga in file Ms-Dos o viceversa (ovviamente sempre in formato Ascii). I programmi di conversione permettono la trascodifica anche dei simboli speciali adattandoli ad uno dei diversi ambienti operativi.

Manualistica e documentazione

Nonostante i manuali siano in inglese, bisogna riconoscere che la Commodore non ha economizzato nel dare all'utente tutte le possibili informazioni in modo chiaro che possano aiutarlo a sfruttare al meglio le caratteristiche della scheda. Molto curata è la descrizione della fase d'installazione e dell'uso delle caratteristiche peculiari della scheda, fra i quali per esempio la creazione di dischi virtuali.

Molto curata anche la parte relativa alla trascodifica dei file fra Amiga e l'ambiente Ms-Dos.

A differenza dei suoi predecessori, questo accessorio di Amiga per emulare l'ambiente Ms-Dos merita un giudizio positivo.

Rimane come sempre il dubbio circa la validità economica di

un'offerta di questo tipo. Come si è già precisato, il costo di una scheda di emulazione corrisponde a quello di un AT compatibile ai prezzi correnti.

Certo la possibilità di avere a disposizione più ambienti contemporaneamente gioca a favore di questa scheda, ma quanti possono apprezzare questa sofisticata possibilità?

Amiga con sprint

La velocità di elaborazione di un computer è determinata da diversi fattori; fra questi il tipo di architettura utilizzato (8, 16, 32 bit) e la conseguente velocità nel trasferimento e nell'elaborazione dei dati. Il processore di cui Amiga è dotato di serie è il famoso Motorola 68000, un tradizionale 16 bit.

La scheda A2620 utilizza invece un processore 68020, un 32 bit, che va a sostituirsi nell'elaborazione dei dati al 68000, trasformando così Amiga in una macchina a 32 bit.

In pratica la scheda è un computer aggiuntivo che si sostituisce al tradizionale Amiga, trasformandolo in un Amiga super-veloce, dotato altresì di unità di gestione a virgola mobile, 68881 e unità di gestione della memoria 68851.

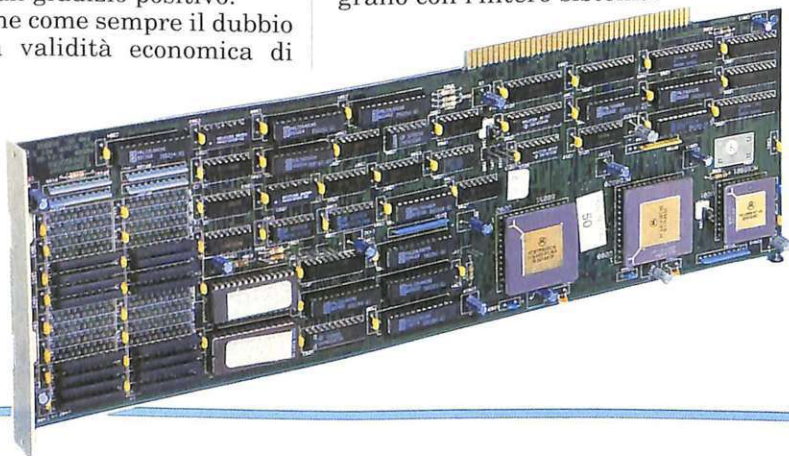
A ciò contribuiscono anche i due Mb di RAM veloce disponibili sulla scheda, che ben si integrano con l'intero sistema.

Installazione

L'installazione della scheda A2620 è relativamente semplice, ma può porre dei problemi a causa delle diverse versioni di Amiga, dovute soprattutto ad alcune variazioni e modifiche avvenute in tempi o luoghi diversi di produzione. È pertanto consigliabile, per evitare danni al sistema, fare installare la scheda a personale specializzato. Nonostante sia sufficiente, in teoria, inserire la scheda acceleratrice nello slot del coprocessore, può in alcuni casi essere richiesta la rimozione del processore principale. Il manuale in dotazione è comunque sufficientemente esauriente circa le modalità d'installazione.

Caratteristiche tecniche

La velocità di elaborazione del processore a 32 bit, pari a 14.3 MHz, consente al processore di eseguire esclusivamente operazioni che non coinvolgano risorse esterne alla scheda; quando deve utilizzare altri dispositivi di Amiga, che lavorano normalmente a 7.2 MHz, il processore non può far altro che adeguarsi a tale fre-





quenza di clock rallentando, tramite opportuni stati di attesa, la propria elaborazione. È evidente come programmi che hanno frequenti operazioni di I/O, o che comunque utilizzano risorse di Amiga, hanno solo modesti incrementi delle prestazioni in conseguenza all'uso della scheda A2-620. I software che richiedono un'elevata elaborazione, quali per esempio quelli per la grafica tecnica o la grafica pittorica, possono invece aumentare notevolmente la propria velocità di esecuzione. Tali programmi accedono solo raramente a risorse esterne, e in più richiedono una rilevante quantità di calcoli.

Costituiscono quindi il classico esempio di software che trae un vantaggio rilevante dalla presenza della scheda A2-620. È chiaro come l'uso di questa scheda sia estremamente legato al tipo di software che il singolo utente utilizza. Nonostante sia pubblicizzato un incremento di prestazioni fino al 400 per cento, la resa è di molto subordinata al tipo di software utilizzato.

Per aumentare ulteriormente le prestazioni del sistema, soprattutto sui programmi sopra citati è inoltre possibile installare un coprocessore matematico più veloce, fino a 33 MHz.

In questo caso si dovrà anche provvedere all'installazione di un opportuno modulo oscillatore della frequenza desiderata. Anche in questo caso l'installazione è relativamente semplice, ma è utile avvalersi di personale altamente qualificato.

Un ulteriore miglioramento si può apportare mediante aggiunta di altra memoria RAM; la scheda è infatti espandibile a quattro Mb (la memoria della scheda viene vista come parte integrante di quella complessiva di Amiga, e pertanto la massima espandibilità del sistema è comunque di otto Mb).

Compatibilità software

La scheda A2620 sostituisce, come già precisato, il 68000 di Amiga con un 68020. È evidente che, sebbene i due processori sia-

no della stessa famiglia, elaborando i dati l'uno a 16 bit e l'altro a 32 bit, possano riscontrarsi delle incompatibilità software, anche a causa della diversa velocità di elaborazione dati.

Proprio per evitare questi inconvenienti il manuale fornisce opportune indicazioni per chi dovrà sviluppare software utilizzando le caratteristiche intrinseche dei processori.

Il software presente sul mercato dovrà essere opportunamente testato per verificarne il corretto funzionamento con l'uso della scheda acceleratrice. In linea di massima tutti i programmi scritti secondo le specifiche elencate nel manuale dovrebbero funzionare correttamente.

La scheda A2-620 è un vero e proprio computer e di questo conserva anche il prezzo. Il suo utilizzo è quindi individuabile in aree professionali come quella della computer graphic.

Infatti solo applicazioni professionali possono giustificare l'investimento richiesto.

Giancarlo Butti

Per trasferire i programmi di RE&C da cassetta a disco

Su ogni numero di Radio Elettronica & Computer trovate l'utility Dsave V2, registrata subito dopo il file d'intestazione. Questo programma consente di trasferire la maggior parte dei programmi della cassetta su un floppy disk. Gli unici programmi che non è possibile trasferire sono quelli che, per ragioni di copyright, sono salvati sulla cassetta in modo autostart.

Il programma Dsave V2 è d'uso semplicissimo: dopo il caricamento del solito menù all'inizio della cassetta si deve caricare e lanciare Dsave v2 con LOAD"DSAVE". Il menù che compare sul video offre tre possibilità:*

1 - La cassetta verrà letta e il primo programma incontrato caricato. A questo punto viene chiesta conferma per il trasferimento sul disco, dopodiché si passerà al caricamento del successivo programma sulla cassetta e così via.

2 - Scegliendo la seconda opzione, invece, verranno salvati su disco tutti i programmi automaticamente, senza selezioni da parte dell'utente. A questo proposito bisogna ricordare che, a volte, sulla cassetta vi sono dei file sequenziali o programmi particolari che il copiatore non può trasferire, nel qual caso il processo si blocca. Vi consigliamo quindi di utilizzare l'opzione 1 anche se volete trasferire tutti i programmi della cassetta.



Le collisioni

Sto lavorando a un programma che utilizza gli sprite, ma ho incontrato un problema che non sono riuscito a risolvere. Come è noto il comando Peek(53278) serve per controllare se due o più sprite hanno avuto una collisione, ma come è possibile sapere tra quali sprite tale collisione è avvenuta?

Roberto Mancini
Pesaro

La locazione 53278 è un registro hardware del Video Interface Controller (Vic). Ogni volta che questo chip ridisegna lo schermo, aggiorna il registro di collisione sprite all'indirizzo 53278. Gli otto sprite, numerati da zero a sette, corrispondono agli otto bit, anch'essi numerati da zero a sette. Il numero che viene prelevato dal registro 53278, per mezzo dell'istruzione Peek, identifica in modo univoco ogni sprite che ha avuto collisione.

Se gli sprite due e tre e nessun altro, sono coinvolti in una collisione, quando effettua una Peek (53278) le verrà restituito il valore 12, che rappresenta la somma dei valori corrispondenti agli sprite coinvolti: quattro per lo sprite numero due, e otto per lo sprite numero tre. Se la collisione avviene tra lo sprite zero e sei, il numero restituito sarà 65. Va tenuto presente che una collisione viene registrata solo quando si sovrappongono i cosiddetti colori di foreground, non quelli di background. Naturalmente è possibile impostare il colore di foreground uguale allo sfondo dello schermo, rendendo così lo sprite invisibile, ma il registro di collisione funzionerà ugualmente.

Anche quando vengono utilizzati gli sprite multicolor, due colori vengono considerati foreground (collisioni verificate), due

invece background. In ogni caso i soli sprite che possono collidere sono quelli accesi sullo schermo, che sono controllati dal registro 53269. Anche se due sprite sono accesi, ma posti fuori dalla parte visibile dello schermo, il registro delle collisioni funzionerà normalmente. I programmatori hanno sempre problemi con le collisioni tra sprite.

Per esempio, diciamo che in un certo momento lo sprite numero zero collide con lo sprite numero uno, e che contemporaneamente, in qualche altra zona dello schermo anche gli sprite tre e sette collidono tra loro.

Leggendo la locazione 53278 si ottiene il valore 139 dato dalla somma dei valori di tutti gli sprite che hanno avuto una qualche collisione ($1+2+8+128$). Tuttavia questo valore dice solo che quattro sprite sono stati coinvolti in una collisione, ma non dice nulla sul numero delle collisioni e tra quali sprite si è verificata. Infatti se fossimo interessati alla situazione dello sprite zero, non potremmo assolutamente sapere se ha colliso con lo sprite uno, tre o sette, o addirittura con tutti.

Una soluzione potrebbe essere quella di spegnere alternativamente i vari sprite coinvolti e controllare continuamente il registro di collisione, sapendo che uno sprite spento non influenza le collisioni.

Lo svantaggio di questo metodo è che spegnendo e accendendo si ha il cosiddetto flickering, cioè una specie di tremolio delle immagini sullo schermo. Inoltre eseguendo una Peek(53278), il valore del registro viene pulito e occorre aspettare un 60esimo di secondo (il tempo di una nuova scansione) prima di rileggerlo.

Una seconda soluzione potrebbe essere di leggere la posizione X e Y degli sprite coinvolti e dedur-

re dai valori delle rispettive coordinate, se sono o meno coinvolti nella stessa collisione.

Linee multi statement

Secondo il manuale del C64 è possibile utilizzare più comandi sulla stessa linea di programma separandole con i due punti, tuttavia il seguente programma non funziona correttamente:

```
10 x1=25: x2=25
20 if x1=0 then 100: x2=x2+1
30 print x1; x2
40 stop
```

Quando faccio girare il programma, sia x1 che x2 valgono 25, invece x2 dovrebbe valere 50. Potreste spiegarmi come mai?

Marcello Bono
Milano

La linea 20 contiene il bug. Un'istruzione If-Then controlla una condizione per vedere se è vera o falsa. In questo esempio la condizione testata è $x1=0$. Se la condizione è vera il comando che segue il Then viene eseguito, più ogni altro comando che segue sulla stessa linea.

Tuttavia, una condizione falsa passa l'esecuzione del programma alla linea successiva. La seconda parte della linea 20, che contiene l'istruzione di assegnamento $x2=x2+x1$ non potrà mai essere eseguita. Infatti se x1 vale zero il programma salta alla linea 100 prima di incontrare l'assegnamento sulla stessa linea, ma se x1 è diversa da zero l'esecuzione passa alla linea 30. Questa caratteristica dell'istruzione If-Then consente di sviluppare programmi che eseguono gruppi di istruzioni diverse in base ad una sola condizione. Occorre ricordare che un If-Then controlla ogni cosa sulla linea corrente.

Il mensile con disco programmi per C64 e C128

COMMO DISK

Anno III - Ottobre 1989 - N. 35 - L. 13.000

ELETTRONICA

**Progetto
di circuiti
al computer**

BASIC

**Effetto fizzling
per la tua grafica**

GRAFICA

**Tu scrivi,
lui disegna**

DRIVE

**Non solo
copiatori**

GIOCO

**Rimbalza
...o muori!**



è in edicola

DISTRIBUZIONE

ocean